What recommendations can be given to IT Managers for using a Continuous Delivery (CD) Maturity Model to implement CD in their organizations?

The focus is on organizational and technical recommendations obtained via analysis of Continuous Delivery Maturity Models, scientific literature and empirical research.
“Would you tell me, please, which way I ought to go from here?”
“That depends a good deal on where you want to get to.” said the Cat.
“I don’t much care where...” said Alice.
“Then it doesn’t matter which way you go.”

— Lewis Carrol, *Alice in Wonderland* —
Preface

One of the best advice I received for study, personal development and successful career, is to make sure you are following your own interests in what you really like to do in life. Only then you are prepared for the tougher situations that will occur, and if your intrinsic driver is already fired up, who can stop you?

The need to continuously improve and deliver is a law of (business) evolution and time to market and customer demands are becoming more challenging in the market place.

IT Managers (like myself) need to give the right example and lead and coach their groups of colleagues and all internal and external stakeholders. Enthusiasm, autonomy, mastery and purpose on the work floor contributes to set the right context and “smell of the place”.

I want to thank my first thesis supervisor Christoph Bockisch and second thesis supervisor Marko van Eekelen for their time, insights and advice. And the much feedback and correction proposals they gave to me, that resulted in a well-guided thesis journey and improved my personal development and the end result of this project.

I want to thank my wife Linda and my kids Kas, Ben and Noor for their understanding me being unavailable for most of the time for half a year, especially in the weekends and holiday periods. I will try to make up for it as the summer 2015 holiday is about to start! And I want to thank my parents for the ability to study in their office every week. I spent a lot of time at their premises and they were always supporting (including delicious food and hospitality). It really contributed to fully concentrate on the research. I want to thank my manager Peter and my peers for their understanding and support. As of now I can spend more time with my family and at work.

Also I would like to thank the contributors to this thesis and interviews: Andreas, Anurag, Craig, Eduard, Lucian, Stijn, Tim and Timme.

And I wish my IT Management peers across organizations a lot of success and hope they also enjoy the Continuous Delivery journey.
Summary

Organizations struggle to deliver the decreasing time-to-market and increased customer demands. Software is key and software engineering and software management processes have significantly improved over time and Continuous Delivery is the new paradigm.

This research tries to contribute with recommendations towards IT Managers for using a Continuous Delivery (CD) Maturity Model to implement CD in their organizations.

This research is executed as a final graduation assignment of the Master of Business Process Management & IT of the Open University of the Netherlands.

Firstly an extensive literature review on this actual topic was performed. Many research questions on the topics of Continuous Delivery, Maturity Models, and how to design, further develop and assess these models are being answered and first recommendations listed.

Secondly an empirical research with semi-structured interviews with Continuous Delivery subject matter experts and IT Managers was performed, to extract data from practice on the added value and usage of Continuous Delivery Maturity Models in organizations.

Continuous Delivery Maturity Models are concluded helpful to better understand throughout an organization the status of CD capabilities and what are the next steps for improvement.

Using CD Maturity models in a Plan-Do-Check-Act cycle approach, next to other tools, has been broadly considered as a best practice.

The organization and its culture are probably the most important aspects to consider when aiming at creating a sustainable Continuous Delivery environment.

IT Management should embrace these developments and opportunity. Planning and prioritization for Continuous Delivery needs to come together into one clear governance and the many benefits of the incremental improvements will start the fire that is needed for this paradigm change!
# Table of Content

1. Introduction and background .................................................................................. 5

2. Designing an empirical study of usefulness of Continuous Delivery ...................... 6
   2.1 Problem definition .............................................................................................. 6
   2.2 Objective description ......................................................................................... 6
   2.3 Conceptual research model ................................................................................ 7

3. Continuous Delivery – The Theory ........................................................................... 9
   3.1 What is Continuous Delivery? ........................................................................... 9
   3.2 What is a Maturity Model? ................................................................................ 11
   3.3 What is a Continuous Delivery Maturity Model? .............................................. 12

4. Continuous Delivery – Scientific Literature Review ............................................... 14
   4.1 Recommendations to IT Managers .................................................................... 14
   4.2 Theoretical framework ....................................................................................... 14
   4.3 Sources ............................................................................................................... 14
   4.4 Search strategy .................................................................................................... 16
   4.5 Results of Continuous Delivery and Maturity Models theory .......................... 17
       4.5.1 Value of a Maturity Model ......................................................................... 17
       4.5.2 Development of a Maturity Model ............................................................. 18
       4.5.3 Assessment of a Maturity Model ............................................................... 20
       4.5.4 Similarities and differences between CD Maturity Models ..................... 23
   4.6 Other recommendations for improving the maturity of Continuous Delivery ...... 24
   4.7 Results of literature review ................................................................................. 25

   5.1 Technical research design ................................................................................... 27
   5.2 Required data and data sources ....................................................................... 27
   5.3 Research strategy ............................................................................................... 28
   5.4 Methods and techniques of data collection ....................................................... 28
   5.5 Threats to validity .............................................................................................. 30
   5.6 Methods of analysis ......................................................................................... 31
   5.7 Questionnaire and interview results ................................................................. 32
       5.7.1 Importance of Continuous Delivery ......................................................... 33
       5.7.2 Value of a Continuous Delivery Maturity Model .................................... 33
1 Introduction and background

The importance of IT and the ability to adopt and change for small, medium and large organizations is constantly growing and it has become a global necessity to stay in business (Smith & Fingar, 2003). Overall, users demand flawless IT solutions, always on and easy to use.

However there is a gap (McConnell, 2002) between best practices for software engineering and software management in theory and what is actually implemented in practice. As a result, organizations are experiencing only a portion of the advantages of the many developments and enhanced standards in software engineering and software management methods of the last 30 years.

In the last decade methods like Agile/SCRUM (Beck et al., 2001), working in small DevOps teams and especially Continuous Delivery have become popular. Iterative software management and engineering methods are considered the most successful new way of IT development to support business processes. However transition is in progress and the required and expected quality and efficiency gains are still not (fully) achieved. The lower than expected success rate of software and project delivery (McConnell, 2002; Thomas, 2002) is leading to disappointments among employees, management and other stakeholders. In particular for sensitive business processes which require high quality, high availability and secure applications, implementing best practices and market standards effectively and efficiently is of utmost importance. Continuous delivery is more than just a new delivery methodology. It is a whole new paradigm for running a business that depends on software (Chen, 2014; Fowler, 2014; Humble & Farley, 2010).

Research executed in the past has discovered possible causes for the gap between theory and practice (Dutta, Van Wassenhove, & Kulandaianwamy, 1998; McConnell, 2002; Thomas, 2002) and there are many written recommendations, usually in the form of best practices. For example the lack of training, knowledge and craftsmanship (McBreen, 2001) of software managers and engineers, as well as lack of automation and other contextual and culture issues are more and more recognized as key missing elements. These missing elements are hindering change and continuous improvements into organizations. Many organizations and IT Managers can clearly explain why change and improvements are needed and most of them can explain what the new way of work is. But the majority of companies lacks in creating a thorough understanding and execution on all levels of the organization how to really implement the preferred iterative way of work and Continuous Delivery and transform successfully. For example a step by step approach, according to iterative principles and continuous improvements is often missing. Steps in the Plan-do-Check-Act Cycle (Deming, 1986) are not executed and Value Stream Mapping (Rother & Shook, 2009) is often forgotten. There are inconsistencies between the way organizations want to function in a future state and how they initiate and act during the transition to this desired state. IT managers play a key role to have this situation improved.
2 Designing an empirical study of usefulness of Continuous Delivery

2.1 Problem definition

The problem definition consists of the objective description and the main and sub-questions for this study and is further explained in the two next subparagraphs. Criteria from literature on how to define a problem statement was used to validate the problem definition, objective description and the final research questions (Gelderman & Ghijsen, 2011).

2.2 Objective description

This research aims at contributing by defining multiple recommendations to IT Managers for using a Continuous Delivery Maturity Model to implement the “best practices” of Continuous Delivery effectively and efficiently into their organizations. The source for these recommendations is found in theory and practice. The focus is on organizational and technical elements and the research will be executed in the context of dedicated DevOps teams working Agile/SCRUM on software development and maintenance of critical application environments. This is similar to the context of the organization I work in myself and the same organization will be used to perform the empirical research.

In total four Continuous Delivery Maturity Models (Benefield, 2010; Humble & Farley, 2010; Minick & Fredricks, 2011; Rehn, Palmborg, & Boström, 2013) were found during the research. They were put in scope for this thesis and have been thoroughly analyzed to better understand their added value and relation with the Continuous Delivery improvement journey for IT Managers. These Maturity Models consist of an overview with categories and levels and a separate description of the model itself.

Commercial Continuous Delivery Maturity Models (for example from front running CD consultancy companies like DZone, Forrester, Praqma, ThoughtWorks and Xebia) are decided by me not in scope of this research, as they are not (always) broadly available and accessible for organizations and IT Managers without conditions and their academical suitability can be questioned.
2.3 Conceptual research model

The conceptual research model gives an overview of what subjects will be researched and in what sequence and for what outcome. The conceptual research model is designed with the purpose to facilitate answering the problem definition (objective description and the main and sub research questions).

Figure 1 outlines the research strategy followed in my project consisting of five steps, which I will detail in the following. A textbox in this drawing summarizes the main activities and results of a particular step. The arrow means that the information and results of a step on the left leads to output into the next box on the right.

The first step of my research is about analyzing scientific literature on the subjects of software engineering, software management and organizational change. The following four main categories were selected as being most relevant:

1. Continuous Delivery
2. Maturity Models
3. Continuous Delivery Maturity Models
4. Agile Maturity Models

As a result of the literature review a solid foundation of definitions and explanations and scope was laid for further investigations in theory and practice.

The second step is executing further research on how Maturity Models for software management and engineering can be assessed. Several Continuous Delivery Maturity Models were analyzed and the similarities and differences between these models were determined. The result of this step are interim conclusions and recommendations.

The third step was to perform empirical research via qualitative analysis with 2 different groups of interviewees: Continuous Delivery subject matter experts (CD Experts and CD Coaches) and IT Managers.

The fourth step was examining and consolidating the analysis results from all the interviews.

And finally the fifth and last step was to draft and present the final conclusions and recommendations.
The research will be carried out comparable to earlier scientific studies. For the assessment of Continuous Delivery Models the approach of Kohlegger (Kohlegger, Maier, & Thalmann, 2009) is used. This is in my perspective an appropriate method to assess Maturity Models on generic criteria. And to present similarities and differences in these maturity models the definitions and categorization of Özcan-Top (Özcan-Top & Demirörs, 2013) has been chosen as fit for purpose.

I follow in my research the same strategy of a previous study on a Business & IT Alignment Maturity Model (Silvius, 2007). The aim of this study is to give recommendations in what way IT can better support business strategies and processes. I apply a similar approach by first starting with an in-depth analysis of the application of a Maturity Model. Then I perform an assessment by interviewing subject matter experts. Interviews with IT Managers are executed as the last part of the empirical research. All these 3 activities are performed in a similar way and sequence as in the research of Silvius.
3 Continuous Delivery – The Theory

3.1 What is Continuous Delivery?

Continuous Delivery (CD) is a software engineering approach in which teams keep producing valuable software in short cycles and ensure that the software can be reliably released at any time (Chen, 2014; Humble & Farley, 2010). It is used in software development to automate and improve the process of software delivery. Techniques such as configuration management, automated testing and deployment and continuous integration (CI) allow software to be developed to a high standard and easily packaged and deployed to test environments. Continuous Integration is a software development practice where members of a team integrate their work frequently – usually each person integrates at least daily, leading to multiple integrations per day. Each integration is verified by an automated build (including test) to detect integration errors as quickly as possible. Many teams find that this approach leads to significantly reduced integration problems and allows a team to develop cohesive software more rapidly (Fowler, 2014). This is resulting in the ability to rapidly, reliably and repeatedly push out enhancements and bug fixes to customers at low risk and with minimal manual overhead.

Getting software released to users is often a painful, risky, and time-consuming process. The authors of Continuous delivery: reliable software releases through build, test, and deployment automation (Humble & Farley, 2010) are recognized in the community for the best theoretical description of this new paradigm. Their ground-breaking approach sets out the principles and technical practices that enable rapid, incremental delivery of high quality, valuable new functionality to users. Through automation of the build, deployment, and testing process, and improved collaboration between developers, testers, and operations, delivery teams can get changes released in a matter of hours, sometimes even minutes, no matter what the size of a project or the complexity of its code base. Jez Humble and David Farley begin by presenting the foundations of a rapid, reliable, low-risk delivery process. Next, they introduce the “deployment pipeline,” an automated process for managing all changes, from check-in to release. Finally, they discuss the “ecosystem” needed to support continuous delivery, from infrastructure, data and configuration management to governance.

The organization and its culture are probably the most important aspects to consider when aiming to create a sustainable Continuous Delivery environment that takes advantage of all the resulting effects (Rehn et al., 2013).

Tests automation, strong team collaboration, effective configuration management, deployment automation and good team culture (Humble & Farley, 2010) are the major practices advocated in CD to boost the effectiveness of a frequent delivery process (Akerele, Ramachandran, & Dixon, 2014). The following aspects have been identified as critical for agile software development practices ( Patel & Ramachandran, 2009):

1. team size
2. client on site
3. team location
While there are a few organizations advocating the need of having the above fully in place from a people, process and technology perspective, most organizations are still maturing along the Continuous Delivery ladder and many have a long way to go. With my research I want to give recommendations to assist these organizations and the IT Managers involved and to experience the many benefits of this new way of work.
3.2 What is a Maturity Model?

Maturing and Maturity Models have been used for decades as an analytic, explanatory or normative concept in several domains, the most well-known of which is software engineering, e.g., Nolan’s stage theory (Nolan, 1973), the capability maturity model integration (CMMI, http://www.sei.cmu.edu/cmmi/) or the Software Process Improvement and Capability Determination model (SPICE, http://www.isospice.com/).

There are many definitions available in the literature for Maturity Models and more specific explanations and applications consist for Maturity Models in multiple scientific domains. The definition for a Maturity Model that I like most from a coverage and completeness perspective is the following by Kohlegger:

*A maturity model conceptually represents phases of increasing quantitative or qualitative capability changes of a maturing element in order to assess its advances with respect to defined focus areas (Kohlegger et al., 2009).*

One of the first models is the Quality Management Maturity Grid (QMMG) and can be described as an organizational maturity matrix conceived by Philip B. Crosby first published in his book *Quality is Free* in 1979 (Crosby, 1980). The QMMG is used by a business or organization as a benchmark of how mature their processes are, and how well they are embedded in their culture, with respect to service or product quality management.

The QMMG is credited with being the precursor maturity model for the Capability Maturity Model (CMM) created a decade later and that also has five levels of maturity. The Quality Management Maturity Grid describes 5 maturity levels through which an organization or business will go through:

1. Uncertainty
2. Awakening
3. Enlightenment
4. Wisdom
5. Certainty

The Software Engineering Institute's (SEI) Capability Maturity Model for Software (CMM) and the International Standards Organization's ISO/IEC 15504 standard for Software Process Improvement and Capability Determination (SPICE) are two important models for software process assessment, improvement and capability determination. The two models have different architecture and focus. SPICE separates processes and capability levels in two dimensions while CMM handles them in one dimension. CMM focuses on an organization’s capability whereas SPICE focuses on single process capability (Varkoi & Makinen, 1998).

The Software Engineering Institute (in cooperation with Carnegie Mellon University) has published relevant articles on the history, evolution and applications for maturity models (Caralli, Knight, & Montgomery, 2012). They conclude that maturity models can help by providing a benchmark to use when assessing how a set of characteristics has evolved over time.
The purpose of a maturity model and the resulting maturity assessment can be divided into three types of maturity models:

1. A descriptive model: no provision for improving maturity or providing relationships to performance. This type of model is good for assessing the here-and-now i.e. the as-is situation.
2. A prescriptive model: provides emphasis on the domain relationships to business performance and indicates how to approach maturity improvement in order to positively affect business value i.e. enables the development of a roadmap for improvement.
3. A comparative model: enables benchmarking across industries or regions. A model of this nature would be able to compare similar practices across organizations in order to benchmark maturity within disparate industries (De Bruin, Freeze, Kaulkarni, & Rosemann, 2005).

Software process capability/maturity models (SPCMMs) can be defined as models that describe best practices for software life-cycle processes, based on good engineering and process-management principles, and process-attribute sets for capability/maturity design aspects (Gresse von Wangenheim et al., 2010). Maturity models should consist of software development and system engineering practices as well as organizational aspects and supporting processes (Schweigert, Vohwinkel, Korsaa, Nevalainen, & Biro, 2014).

A maturity model consists of a sequence of maturity levels for a class of objects. It represents an anticipated, desired, or typical evolution path of these objects shaped as discrete stages (Becker, Knackstedt, & Pöppelbuß, 2009). The levels are not strict and mandatory stages that need to be passed in sequence, but rather should serve as a basis for evaluation and planning. It is however important to try to keep the overall maturity level fairly even and to keep in mind that big changes may cause skepticism and reluctance in the organization, so an incremental approach to moving through the levels is recommended. And each category has its own maturity progression but typically an organization will gradually mature over several categories rather than just one or two since they are connected and will affect each other to a certain extent (Rehn et al., 2013) (Minick & Fredricks, 2011) (Humble & Farley, 2010).

### 3.3 What is a Continuous Delivery Maturity Model?

A Continuous Delivery Maturity Model is a maturity model grid with descriptions (levels and categories) of characteristics, attributes, indicators, and patterns associated with Continuous Delivery.

A research by Kohlegger and others identified 74 different maturity models from domains within the spectrum of (business) information systems and computer science and shows the great variety and widespread use of maturity models available in the literature (Kohlegger et al., 2009). However Continuous Delivery Maturity Models are a new phenomenon and scientific information is scarce.
To obtain insights, an internet search was undertaken as part of this thesis. At the time of writing this thesis no commonly accepted and widely used Continuous Delivery Maturity Model was available.

Continuous Delivery is a relatively new paradigm and this explains the reason for the paucity of research work done in this field (Akerele et al., 2014). Maturity Models should evolve over time and re-use earlier versions of Maturity Models and iteratively should include the prosed improvements of the user community. I have found the following Continuous Delivery Maturity Models and they are presented in chronological order below:

1. The Continuous Delivery Maturity Model for Continuous Delivery Configuration and Release Management (Humble & Farley, 2010).
2. The Continuous Delivery Maturity Assessment Model (Benefield, 2010).
3. The Enterprise Continuous Delivery Maturity Model (Minick & Fredricks, 2011).
4. The Continuous Delivery Maturity Model (Rehn et al., 2013).
4 Continuous Delivery – Scientific Literature Review

4.1 Recommendations to IT Managers

The following main research question has been derived from the objective description:

*What recommendations can be given to IT Managers for using a Continuous Delivery (CD) Maturity Model to implement CD in their organizations?*

As this is a (too) broad and generic research question, I decided to break it down in smaller sub questions. The goal of the scientific literature review is to find solid answers in referenced literature to these sub questions. An interim set of conclusions and recommendations is drafted after answering the sub questions (see step 2 of the conceptual research model in Figure 1) and concludes the literature review.

4.2 Theoretical framework

A detailed description of execution of the literature study will be reported in the coming paragraphs and the different approaches and choices will be further explained. First the main research question and sub questions were defined and listed as its key to understand which exact information needs to be found in literature.

4.3 Sources

Sources for this literature review are scientific publications and articles, conference proceedings, interviews, lessons learned and questionnaires published in magazines or on the Internet from experts in the global software engineering and software management community. Also the Continuous Delivery Maturity Model described in the book: “Continuous Delivery: Reliable Software Releases through Build, Test and Deployment Automation” (Humble & Farley, 2010) is referred to in this research. Endnote X7 is used from the start of the literature study to automatically manage all references according APA 6th style (see also chapter “References”).

On Wikipedia (Wikipedia, 2015) the list for the academic databases and search engines for the Computer Science scientific domain is retrieved and a selection is made. Also advice on usage of specific academic databases and search engines from both the thesis supervisors is taken into account. In the overview of Figure 2 the final selection is presented. All these sources are being used to find relevant information for the literature review. I used this approach after reading an article on how to conduct an effective literature review in support of information systems research (Levy & Ellis, 2006).
<table>
<thead>
<tr>
<th>Name</th>
<th>Discipline</th>
<th>Description</th>
<th>Access costs</th>
<th>Provider</th>
</tr>
</thead>
<tbody>
<tr>
<td>CiteSeerX</td>
<td>Computer Science, Statistics, Mathematics, becoming Multidisciplinary</td>
<td>CiteSeerX is an evolving scientific literature digital library and search engine that has focused primarily on the literature in computer and information science.</td>
<td>Free</td>
<td>Pennsylvania State University</td>
</tr>
<tr>
<td>DBLP</td>
<td>Computer Science</td>
<td>Comprehensive list of papers from major computer science conferences and journals</td>
<td>Free</td>
<td>University of Trier, Germany</td>
</tr>
<tr>
<td>Google</td>
<td>Computer Science and other domains</td>
<td>Provides possibility to explore scientific papers, conferences, journals, and authors.</td>
<td>Free</td>
<td>Google</td>
</tr>
<tr>
<td>IEEE Xplore</td>
<td>Computer Science, Engineering, Electronics</td>
<td>Library of articles published by IEEE.</td>
<td>Subscription via OU Digital Library</td>
<td>IEEE</td>
</tr>
<tr>
<td>OU Digital Library</td>
<td>Computer Science and other domains</td>
<td>Digital Library of the Open Universiteit in cooperation with Maastricht University with access to many information files and libraries.</td>
<td>Subscription</td>
<td>Open Universiteit</td>
</tr>
<tr>
<td>SpringerLink</td>
<td>Computer Science and other domains</td>
<td>SpringerLink is one of the world's leading interactive databases for high-quality STM (Science, Technology, Medicine) journals, book series, books, reference works</td>
<td>Free</td>
<td>SpringerLink</td>
</tr>
</tbody>
</table>

Figure 2: List of sources to find scientific articles, papers from major computer science conferences and journal articles.
4.4 Search strategy

The first approach to find relevant scientific articles, papers from major computer science conferences and journals was to start searching on key words via the following search engines (see Figure 2):

1. OU Digital Library (http://bibliotheek.ou.nl)
2. DBLP Computer Science Bibliography (http://dblp.org/search)
3. CiteSeerX (http://citeseerx.ist.psu.edu/index)
5. SpringerLink (http://link.springer.com)
6. Google (https://www.google.nl)

A structured search was performed by using the same keywords and selections for all used platforms. The keywords that were used were: Continuous Delivery, Continuous Delivery Maturity Model, Continuous Delivery Maturity, Continuous Delivery Model, Continuous Delivery Assessment, Continuous Delivery Self-Assessment and Maturity Model.

All publications returned by these search criteria have been assessed whether the articles could contribute sufficiently to answer the research questions.

If possible the search domain was refined towards Computer Science to increase the chance on positive findings. If too many search results were presented (more than 500 results to manually scan) then extra refinement was used, depending on the possibilities of the search engine. Examples are:

- using only relevant fields of expertise (Computer Science or Software Engineering).
- setting a minimum number of citations (minimum of 10 citations).
- limiting the publishing date (article not older than year 2010).
- strengthening keyword search based on AND statements.
- searching articles by name of author(s).

In total 30 articles were selected to be relevant for answering (part of) the research sub-questions. After reading them thoroughly, eight articles were determined to be less relevant after all and excluded. For example I decided to out scope an article on Rugby (Krusche, Alperowitz, Bruegge, & Wagner, 2014) as this research is done on part-time developers only and that is not consistent with the scope and practice organization that was defined (mostly consist of full-time developers). And seven articles were added after being referred to in other relevant articles. Many of the articles come up as positive result on several search engines. To avoid doubles a reference registration was consequently maintained via the Endnote reference manager. A complete overview is administrated separately on which article is found via which search engine and based on what search key words and/or other criteria and can be requested to the author of this thesis. Several searching methods like snowball were used. By using the snowball method, articles have been found to which other writers referred.
4.5 Results of Continuous Delivery and Maturity Models theory

Goal of this section is to identify a basis for comparing CD maturity models and for estimating their potential value for a company. For this purpose, I will first outline the value of maturity models in general (Section 4.5.1). In Section 4.5.2, I discuss requirements for maturity models which form a foundation of assessment methods for maturity models, discussed in Section 4.5.3. There I also present my application of these general assessment methods on continuous delivery maturity models in particular. In Section 4.5.4 the analysis of similarities and differences of maturity models is described and in Section 4.5.5 further recommendations to improve Continuous Delivery Maturity Models are presented.

4.5.1 Value of a Maturity Model

The main value of a maturity model lies in its use as an analysis and positioning tool. The framework and carefully developed set of criteria are invaluable to organizations with an urgent and persistent need to understand where they are in relation to the externally determined “best practices” of today. Maturity models are designed to help organizations recognize when and why they should move forward and to provide them with an insight into what action they need to take in order to advance (Duffy, 2001). And they help to identify where an organization stands in terms of the maturity of its processes and practices and defines a progression that an organization can work through to improve (Humble & Farley, 2010). This is to give structure and understanding to the implementation of Continuous delivery and its core components (Rehn et al., 2013). Maturity models can give a starting point and a base for planning the transformation of the company towards Continuous Delivery (Rehn et al., 2013).

Considering an (agile) maturity model is of high interest and is something of relevance for an organization (Schweigert et al., 2014). Maturity models are popular instruments used, e.g., to rate capabilities of maturing elements and select appropriate actions to take the elements to a higher level of maturity (Kohlegger et al., 2009). With maturity models representing theories of stage-based evolution, their basic purpose consists in describing stages and maturation paths (Pöppelbuß & Röglinger, 2011). Using a Maturity Model can answer how mature the organization is in Continuous Delivery and automation practices. And it can help to identify where they can get the most improvement for their specific problems and needs (Minick & Fredricks, 2011). Maturity models are available for evaluating and comparing process improvements or assessments, based on the assumption that higher process capability or organizational maturity are associated with better performance (Gresse von Wangenheim et al., 2010).

In practice, the overall adoption of maturity models is expected to increase, a prediction corroborated by the numerous proprietary models proposed by software companies and consultancies. Recent literature also reports an increasing academic interest in maturity models (Pöppelbuß & Röglinger, 2011). In IT management, maturity models have proved to be an important instrument because they allow for a better positioning of the organization and help finding better solutions for implementing process improvements. Over the last few
years, over a hundred maturity models have been developed to support IT management (Becker et al., 2009). IT management needs supportive tools to assess the as-is situation of a company, derive and prioritize improvement measures and subsequently control the progress of their implementation (Becker et al., 2009). Understanding and mapping the maturity of practices for interdependent teams and units provides a method to discover and remove bottlenecks between groups that enable the organization to continuously improve. For example, Benefield researched the use of a CD Maturity Model within British Telecom. As a result seven practices, or dimensions, have been uncovered during the team experiments, which led to significant quality and velocity improvements across the solution stack when they are used together (Benefield, 2010).

Using Maturity Models can create and evolve a common language within an organization. Maturity Models often create a consistent way of thinking and communicating about that domain that is embodied in the maturity model language or taxonomy. Consistent language and communication supports domains of knowledge to evolve into disciplines where a common language can translate into repeatable, consistent, and predictable performance improvements over time (Caralli et al., 2012).

Overall we can conclude that there is significant value for using Continuous Delivery Maturity Models and that it is recognized in the industry for example because more organizations using these models.

4.5.2 Development of a Maturity Model

The proposed standard development framework is: Scope → Design → Populate → Test → Deploy → Maintain. This clear sequencing should form a sound basis to iteratively guide the development of a model. First development through the descriptive phase, and then evolution of the model through the prescriptive and comparative phases (De Bruin et al., 2005). There are requirements for the development of maturity models available. The main purpose of the 8-step procedure model proposed below is to raise awareness for a methodologically well-founded maturity model design (Becker et al., 2009):

1. **Comparison with existing maturity models**: The need for the development of a new maturity model must be substantiated by a comparison with existing models.
2. **Iterative Procedure**: Maturity models must be developed iteratively, i.e., step by step.
3. **Evaluation**: All principles and premises for the development of a maturity model, as well as usefulness, quality and effectiveness of the artifact, must be evaluated iteratively.
4. **Multi-methodological Procedure**: The development of maturity models employs a variety of research methods, the use of which needs to be well-founded and finely attuned.
5. **Identification of Problem Relevance**: The relevance of the problem solution proposed by the projected maturity model for researchers and/or practitioners must be demonstrated.
6. **Problem Definition**: The prospective application domain of the maturity model, as well as the conditions for its application and the intended benefits, must be determined prior to design.
7. **Targeted Presentation of Results:** The presentation of the maturity model must be targeted with regard to the conditions of its application and the needs of its users.

8. **Scientific Documentation:** The design process of the maturity model needs to be documented in detail, considering each step of the process, the parties involved, the applied methods, and the results.

Although there are many maturity models reported in scientific and non-scientific literature, the act of how to develop a maturity model is for the most part unexplored. Many maturity models simply build on their, often well-known, predecessors without critical discourse about the appropriateness of their underlying assumptions (Kohlegger et al., 2009). The results are transformed into a set of questions which can be used for the (re-)creation of maturity models (Kohlegger et al., 2009). The key question is which design principles are helpful to make a maturity model useful for its intended application domain and purpose of use. According to Pöppelbuß et al., there are no such design principles and no corresponding classifications as yet. They propose general design principals for maturity models based on an extensive review of maturity-model-related literature. To categorize the usefulness of maturity models, design principles can be grouped into basic principles, principles for a descriptive purpose of use, and principles for a prescriptive purpose of use (see also the different types of maturity models listed in the maturity model chapter) (Pöppelbuß & Röglinger, 2011).

A draft maturity model is seldom validated systematically before publication. And when the model is validated, it’s usually through an expert review with varying degrees of participation (Gresse von Wangenheim et al., 2010).

Proença et al. conclude that Maturity Models are often developed ad hoc, without following a well-documented design and development method, and often do not provide a pathway to further extend and update the model to foster systematic enhancements and extensions (Proença et al., 2013).

Re-using existing Continuous Delivery Maturity Models and simplify the as-is assessment and the determination of the next steps is considered to be an improvement. If the “grid” is rather similar then this improvement should be feasible to implement.
4.5.3 Assessment of a Maturity Model

Self-assessment is the most common way of performing software process assessment. The popularity for self-assessment lies in its low cost, good accessibility and ownership of the result (Dutta, Lee, & Wassenhove, 1999). The main requirements and constrains are: (Buglione, 2011):

1. The appraisal(s) should be cheap, quick and sufficiently detailed for writing effective improvement plans.
2. The appraisal method(s) should be simple to be understood and produce short reports for management, showing in one slide what is going on and where to intervene for improving results.

A clear definition of the most relevant drivers for improvement should be done and shared across the organization/team(s). But how to assess the assessment tool itself? In Agile Maturity Model (AMM) Assessment, the area of improvement is identified if the answer of the questionnaires is Yes, Partially, No or Not Applicable. Using these criteria the percentage for each key process area (KPA) can be calculated quantitatively (Patel & Ramachandran, 2009). Adoption of the typical 4-level ordinal scale (N/P/L/F) has been adopted by most known models (CMMI; SPICE), even if using a different percentage distribution across levels (Buglione, 2011).

<p>| | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>N</td>
<td>Not Achieved</td>
<td>0-15%</td>
</tr>
<tr>
<td>P</td>
<td>Partly Achieved</td>
<td>16-50%</td>
</tr>
<tr>
<td>L</td>
<td>Largely Achieved</td>
<td>51-85%</td>
</tr>
<tr>
<td>F</td>
<td>Fully Achieved</td>
<td>86-100%</td>
</tr>
</tbody>
</table>

Figure 3: Example of mapping Not/Partially/Largely/Fully Achieved into a percentage score range.

An analysis of the quality of a subset of agile maturity models was performed by Özcan-Top and Demirörs (Özcan-Top & Demirörs, 2013). Instead of asking about the development and evolution history of the model, this analysis took a set of six model quality criteria. These criteria are: fitness for purpose, completeness, definition of agile levels, objectivity, correctness and consistency. They checked if these quality criteria were fully, largely, partially or not fulfilled by the models. As an overall result, it could be said that the rating of the analyzed subset on Agile Maturity Models was more or less poor.

Similar to the analysis by Özcan-Top, I perform an analysis on the four Continuous Delivery Maturity Models selected in Section 3.3. The result is summarized in Table 1 below. The major purpose of these analyses is to identify and compare weaknesses and strengths of these four CD maturity models and their usability/applicability for assessing the organization’s software development processes. Below, I discuss these results in detail:
**Fitness for Purpose:** A CD maturity model/framework must be developed with the purpose of assessing CD process capabilities and assisting organizations in software process improvement. Among the research models *CD MM 2 Benefield* is developed for usage in one specific organization and the other three CD MMs are developed for generic usage in multiple organizations. Because *CD MM 3 Minick* has much less in scope from a process assessment perspective, it also receives a lower “partly achieved” score.

**Completeness:** A CD maturity model/framework must address all or a subset of major engineering and management processes within a software development life cycle. It must include process related definitions, goals, practices or process success indicators which enable assessment of the CD processes. *CD MM 4 Rehn* is complete as also the culture and organizational aspects and information and reporting capabilities are being assessed. These components are missing in the other three models. *CD MM 3 Minick* is simplified to only four assessment areas and receives the partly achieved score.

**Definition of CD Levels:** A CD maturity model/framework must provide definitions of CD levels which enumerate the different degrees of agility. Those maturity levels need to be interpreted intuitively and must be designed to complement each other. For all four CD MM’s this is largely in place with having 5 levels of maturity defined.

**Objectivity:** At the end of a maturity assessment, verifiable results must be produced. The judgment of the assessor must be at a minimum level. In *CD MM 1 Humble* there is a very big step for the lowest maturity and the next stage. In the other three models there are multiple objective criteria to assess per category. It is not clear if you need to meet at least one criterion in a level or all criteria to reach that particular level. As there is no standard or required assessment execution rule for showing an objective is fully achieved, none of the maturity models achieves the full score on this item.

**Correctness:** All model elements must be compatible with CD principles. Descriptions, goals and work products must correctly represent the related process or process area. I have not found incorrect CD principles and consider all CD MMs as fully correct.

**Consistency:** A CD maturity model/framework must be internally consistent. All processes and practices must be at the same logical level. There must be no logical or temporal conflicts between two specified model elements. Overall I rated the consistency as largely consistent. The fact is that within maturity levels certain components can be reached independently and have added value for the organization. However this is not fully recognizable in a maturity gain in the model, so not fully consistent.
<table>
<thead>
<tr>
<th>Criteria/Models</th>
<th>Fitness for Purpose</th>
<th>Completeness</th>
<th>Definition of the CD Levels</th>
<th>Objectivity</th>
<th>Correctness</th>
<th>Consistency</th>
</tr>
</thead>
<tbody>
<tr>
<td>CD MM 1 Humble</td>
<td>Largely Achieved</td>
<td>Largely Achieved</td>
<td>Largely Achieved</td>
<td>Partly Achieved</td>
<td>Fully Achieved</td>
<td>Largely Achieved</td>
</tr>
<tr>
<td>CD MM 2 Benefield</td>
<td>Partly Achieved</td>
<td>Largely Achieved</td>
<td>Largely Achieved</td>
<td>Largely Achieved</td>
<td>Fully Achieved</td>
<td>Largely Achieved</td>
</tr>
<tr>
<td>CD MM 3 Minick</td>
<td>Partly Achieved</td>
<td>Partly Achieved</td>
<td>Largely Achieved</td>
<td>Largely Achieved</td>
<td>Fully Achieved</td>
<td>Largely Achieved</td>
</tr>
<tr>
<td>CD MM 4 Rehn</td>
<td>Largely Achieved</td>
<td>Fully Achieved</td>
<td>Largely Achieved</td>
<td>Largely Achieved</td>
<td>Fully Achieved</td>
<td>Largely Achieved</td>
</tr>
</tbody>
</table>

Table 1: Qualitative analysis of CD Maturity Models.
4.5.4 Similarities and differences between CD Maturity Models

An analysis of a sample of maturity models has shown that many models differ with respect to their characteristics. But at the same time also several similarities within these models have been found which in part can be explained by the fact that many authors of maturity models simply build on their predecessors without much thinking about the appropriateness of their design decisions. This suggests that the concept of maturity modelling should be reflected and reassessed. Multiple aspects can be used as checklist for selecting a maturity model, for comparing maturity models or for designing maturity models. (Kohlegger et al., 2009). Then also the items that are missing should come to the surface. To answer this sub research question I performed a similar analysis as Kohlegger (based on structured content analysis of Mayring in 2008) to define similarities (all maturity models (MM) score equally on the same criteria) and differences (one or more MMs score differently in the selected criteria) of the four CD Maturity models. The full results are presented in tables 2 and 3 in Appendix A.

The main similarities are that all selected maturity models consists of 5 maturity levels and the meaning of maturity has the same description. And in all models skipping a level is not recommended and there are no supporting tools available.

The main differences between these models are the different categories and descriptions, the availability of a starting stage and if an existing model is used as foundation to draft the updated maturity model.

To make this assessment more objective, the empirical study presented in Section 5 has been carried out. This provides an expert opinion of the similarities and differences and preferences from a practical perspective from CD Experts and IT Managers. This is necessary to conclude which of the characteristics and/or criteria are most important for the purpose of Continuous Delivery Maturity Models and which elements are being preferred over others. The results of the empirical research are presented in Section 6.
4.6 Other recommendations for improving the maturity of Continuous Delivery

Apply the Deming cycle: plan, do, check, act for updating and improving Continuous Delivery maturity in organizations (Humble & Farley, 2010).

1. *Classify Continuous Delivery Maturity:* You may find different levels in each of the different categories.
2. *Choose focus areas where immaturity is painful:* Value stream mapping can assist in further identifying these areas. Prioritize the improvements on their business case rationale. Define acceptance criteria for the results to decide if changes are successfully achieved.
3. *Implement changes according to a plan:* Determine a few goals and describe them in detail. Also describe the activities that should be actioned by defined parties at what time achieve those results.
4. *Measure if the changed have the desired effect:* by using the pre-defined acceptance criteria.
5. *Repeat the above steps:* and roll-out improvements incrementally.

Start to improve that part of the organization where people are open for changes and really willing to cooperate on improvements. This people will have the best motivation to start working on Continuous Delivery improvements and become the advocates for further automation (Humble & Farley, 2010). This people can often be found in an area with ongoing issues and escalations and a low maturity score.

Finding the changes that will deliver the most value and working out how to execute them should be treated scientifically. Come up with a hypothesis, then test, repeat and learn in the process (Humble & Farley, 2010). Define a generic process model for software process improvement that is suitable for agile software development environments, to identify and define agile practices for each maturity level and relate agile practices problem to agile practices improvement goals (Patel & Ramachandran, 2009).

A team without reporting is flying blind. All the testing in the world is useless if no one can review the results. Interesting and adding value metrics can e.g. be cycle-time, delivery time, number of releases, number of emergency fixes, number of incidents, number of features per release, bugs found during integration test etc. (Rehn et al., 2013). When more mature also real-time information, business metrics and trend reporting should be used across the delivery pipeline. Development cycle time, quality and effort are main metrics (Akerele et al., 2014). Likewise reams of data that has not been distilled into digestible information can be so difficult to learn from that they become useless. Maturing teams have defined metrics and dashboards that are actual and visible, and expose increasingly useful information (Minick & Fredricks, 2011). Focus on clear goal, for example reduction of cycle time to support a successful CD implementation (Rehn et al., 2013).

If software engineers had better knowledge about how Software process capability/maturity models (SPCMMs) are developed and the basis of their recommendations, they might be able to interpret and use them to optimize their benefits. These issues were analyzed in a systematic literature review and follow-up questionnaire (Gresse von Wangenheim et al.,
Light Maturity Models (LMM) can be considered a bridge between the non-usage and the full usage of MMs, hoping to provide also for small and medium enterprises (SME) and very small entities (VSE) a cheaper way to overcome the typical efficacy/effectiveness criteria from the ‘audit’ world while moving towards measurable ‘appraisals’. And a plenty of information can arise from process appraisals, creating value when properly addressed with specific corrective/improvement actions. A process appraisal that only asks for a little effort will have a higher chance to be adopted by a small (DevOps) team (Buglione, 2011).

Although literature exists on organizational change, there seems to be very little, if any, research specifically focusing on introducing CD to an organization. Further research on this topic, e.g., understanding the challenges in more depths and developing strategies and practices to tackle them more effectively, will significantly help an organization’s smooth adoption of CD (Chen, 2014).

Challenges that can be encountered when introducing Continuous Delivery comprise resistance from partners and suppliers. They can be mitigated for example by education and Outreach to key stakeholders throughout the business. Commercial Off the Shelf (COTS) products, as well as hardware and process integration, do not integrate into this framework in a straightforward fashion (Benefield, 2010).

IT management plays a key role in the determination of gaps and adopting of the associated recommendations into their organizations.

**4.7 Results of literature review**

IT is extremely important in current business and personal usage and many organizations struggle to have IT delivery improved. Despite the fact that there are many “best practices” available, the reality is that overall maturity can and should be improved. IT Management plays a key role in stimulating, leading and coaching during this transition.

Relevant literature in the domain of Computer Science that is published in the (recent) past has been examined via a structured review approach. Many interesting articles were found via search engines, scientific literature digital libraries and universities and by using referred documentation on all sub questions a detailed answer has been given. In general there does not seem to be much conflict in literature on the main subjects.

However there is currently no commonly accepted and widely used Continuous Delivery Maturity Model. Continuous Delivery is a relatively new paradigm and this explains the reason for the paucity of research work done in this field and I hope my efforts contribute a piece to the developments. I have chosen four broadly available Continuous Delivery Maturity Models and assessed the many similarities and differences.

The main value of a maturity model lies in its use as an analysis and positioning tool. IT management needs supportive tools like a CD maturity model to assess the as-is situation of the organization, derive and prioritize improvement measures and subsequently control the progress of their implementation of further improvements. This is to gradually mature over several categories like automatic deployment, testing and configuration management.
Another goal is to define and implement improvements on non-technical area’s like culture and team collaboration, ownership and removing organizational boundaries.

Many definitions and types of maturity models have been found in the literature. Multiple frameworks for designing and assessing CD Maturity Models have been examined and multiple recommendations from theory and practice have been found. The recommendation from the literature is to focus improvements on parts of the organization where the process quality is low. Value stream mapping can assist in further identifying these potential areas. Improvements should be prioritized on their business case rationale. Acceptance criteria should be defined for the results to decide if changes are successful. Maturing teams have defined metrics and dashboards that are actual and visible, and expose increasingly useful information. Tests automation, strong team collaboration, effective configuration management, deployment automation and good team culture (Humble & Farley, 2010) are the major practices advocated in CD to boost the effectiveness of a frequent delivery process (Akerele et al., 2014). The following aspects have been identified as critical for agile software development practices (Patel & Ramachandran, 2009):

1. team size
2. client on site
3. team location

Patel and Ramachandran recommend to define a generic process model for software process improvement that is suitable for agile software development environments. This is to identify and define agile practices for each maturity level and relate agile practices problem to agile practices improvement goals (Patel & Ramachandran, 2009). Re-using existing Continuous Delivery Maturity Models and simplify the as-is assessment and the determination of the next steps is considered to be an improvement. If the “grid” is rather similar then this improvement should be feasible to implement.

The criticism found in literature is that the intended quality and performance goals are not explicitly stated in the investigated research.

To further validate the many recommendations towards IT managers more empirical research is needed. As part of my thesis I will perform a practical case study in my own organization and the approach, execution and conclusions are presented in Section 5.
5 Continuous Delivery – The Reality – Empirical Research

5.1 Technical research design

The technical research design describes in detail how the empirical research will be executed. It comprises the following components: the required data and data sources, operationalization, the research strategy, methods and techniques of data collection, measurement levels, validity and reliability, methods of analysis and a preview of the results.

5.2 Required data and data sources

The scientific approach, as described in the chapter conceptual research model, is to discuss the research main and sub questions similar to the literature review of theory also in practice.

The first step is to discuss the interim summary and conclusions of the literature study with a group of CD Experts/Coaches. In this discussion the Continuous Delivery Maturity Model of Rehn will be used. I have chosen this maturity model as it achieved the highest score in the assessment described in Section 4.5.3. This group of CD Experts/Coaches validates which pre-defined focus areas are specifically important for Continuous Delivery. And it is expected they will give valuable input and recommendations for improving CD maturity models. And they can potentially sharpen the formulation of the research questions being used in the interview rounds with IT Managers. This outcome of the CD Experts/Coaches discussions is to be used as the “training set” for the empirical research and the follow up interviews with IT Managers.

The second step is to prepare even more specific interview questions towards a diverse group of IT managers. They validate and respond to the questions of CD Maturity Model of Rehn in practice. I request their feedback and recommendations for this qualitative analysis. All data is derived from the interview questions during the interviews in the global organization. The information is collected and analyzed and is to be considered as the “validation set” for the empirical research. The interview and analysis activities are executed by myself. For the interviews I need the cooperation and time of multiple colleagues to receive the information that I need.

It is preferred to obtain broad applicability and generalizable conclusions of my research. Due to time constraints of a master project my research focuses on my own global organization and business sector. The CD Experts/Coaches that are involved in the interviews have knowledge and experience outside of the practice organization. This will give me more possibilities to qualify and generalize the research recommendations and conclusions. To further increase the confidence in the research results, in future work the interviews should be extended to CD experts and IT managers in other organizations and business sectors.
5.3 Research strategy

The research philosophy is in the area of positivism being part of the epistemology (a theory of understanding) container. The research is in the field of software engineering and management science. It should be clear which assumptions I am making and what is considered as obvious. Only then can I investigate and challenge those assumptions and explore the recommendations in more detail. I will mainly follow the approach of the deduction method (Saunders, Lewis, Thornwill, Booij, & Verckens, 2011). This to ensure that I can make strong conclusions on the data received from the empirical research compared with the theory and scientific literature. To collect extra contextual information (using the “Why?” and “How?” questions) I will use the induction method (Saunders et al., 2011).

During the empirical research I will investigate in what way organizations can make use of CD Maturity Models and what aspects are relevant for implementing Continuous Delivery while working Agile/SCRUM in dedicated DevOps teams. For this research I have chosen the single case study method (Saunders et al., 2011) and research will be executed within my own organization to benefit from the easy access to data sources meaning my own colleagues.

The exact data derived from empirical research are words (so no numbers) via semi-structured interviews. These words as answers to the questions asked during the interview will be recorded, fully transcribed, summarized and further analyzed and compared with the output of the scientific literature review.

A Continuous Delivery Expert and Coach is a person that has professional experience for more than 5 years in the function of teaching and coaching individual employees, teams and IT Managers on implementing and improving Continuous Delivery in organizations.

An IT Manager is a person that has professional experience for more than 5 years in the IT function of leading and coaching individual employees and teams. And currently working in the context of dedicated DevOps teams working Agile/SCRUM on software development and maintenance of critical application environments. And has experience for at least 1 year with implementing CD in the organization.

The Commercial Banking IT organization (approximately 1200 employees) is part of a large international Financial Institution. This organization will be used to test and validate the recommendations deducted from scientific literature in practice. As described in Section 2.3 the research will follow the approach of a study on a Business & IT Alignment Maturity Model by Silvius (Silvius, 2007).

5.4 Methods and techniques of data collection

As the empirical research will take place in one organization (Commercial Banking IT) it can be considered as a single case study. The reason is the limited amount of time and effort to execute the research and the benefit of having easy access to data sources available.
However the departments within this extensive and global organization are logical sub elements, so the type of case study is an embedded case study (Saunders et al., 2011). As a consequence the results and recommendations could be generalized to other organizations and companies as well, and further research could be executed to scientifically justify the generalization. Action research (Saunders et al., 2011) has been also identified as an appropriate research method for my thesis, however requirements of the limited duration (and effort) of the empirical research could not be achieved, hence this method has not given the preference over the single case study (embedded).

The type of research can also be described as exploratory research. I want to better understand the gap between theory and practice, ask questions and come up with broader recommendations. I will conduct qualitative research and this shall deliver unstructured information as an outcome. Main goal is to understand which factors play an important role, the causal relationships and root causes of events. Also the social perspective and motivations of individuals will be taken into account. For the exploratory research both semi-structured or not-structured interviews are considered as most appropriate interview techniques (Saunders et al., 2011).

To adapt to the preference of managers in general to be interviewed over filling questionnaires and to achieve a sufficient response rate semi-structured interview will be used. There are many questions (total of 13 questions) and some of them are open and complex questions and I have the possibility (and flexibility) to change order of the questions. There is some overlap in the questions asked, and this will contribute to receive complete information. I will ensure all questions are answered before closing the interview.

For the empirical research the main sources identified are Continuous Delivery Experts/Coaches and IT Managers. They have significant experience in the field of expertise and have knowledge of the different topics and are exposed to the transition to CD way of work (at least 5 years for CD Experts/Coaches and 1 year for IT Managers). To use the structured interview method is an appropriate method for qualitative review (Myers & Newman, 2006). And the result of the interviews is data gathering. The sources for the empirical research that can answer the research (sub) questions will be interviews (Baker & Edwards, 2012). I concluded that if the interviews with IT managers at a certain moment do not release more recommendations already given by other IT Managers, the number of interviews can be considered as sufficient. I conducted 3 interviews with CD experts/coaches (using Continuous Delivery Maturity Models for years in different organizations) and 4 interviews with IT Managers. Two of the IT Managers are working in the Financial Markets domain and the other two IT Managers are working in the Channels and Payments domain. I have not found specific differences caused by the fact that these managers work in different departments in the organization, and thus could not find obvious or mandatory reasons to not generalize the results of the interviews.

In total 7 interviews have been executed, transcribed, analyzed and after combining with the output of the literature review it will result in final conclusions and recommendations. Qualitative analysis tooling will be used when appropriate.
5.5 Threats to validity

As the primary data to be obtained is nonnumeric, it requires a qualitative research to gather this information. And it could be categorized as a cross-sectional study (compared to a longitudinal study) (Saunders et al., 2011) as it focuses on the current moment only and information gathered within a short timeframe. My research can have multiple biases. The one that I consider as most relevant is the participant bias and specifically the participant response bias (Saunders et al., 2011). As the participants are working at the moment of the interviews in the same organization as the researcher (me), it can be the case that they provide (socially) desirable responses during interviews and that may bias the response of the participant in some way. In essence, it is a bias that drives an individual to answer in a way that makes them look more favorable to the experimenter. Therefore, it is important for researchers to employ strategies aimed at mitigating social desirability bias so that they can draw valid conclusions from their research.

Several strategies exist (Saunders et al., 2011) to limit the effect of social desirability bias and I have selected the following:

**Forced-choice items:** This technique hopes to generate questions that are equal in desirability to hopefully prevent a socially desirable response in one direction or another.

**Neutral questions:** The goal of this strategy is to use questions that are rated as neutral by a wide range of raters so that socially desirable responding does not apply.

**Self-administered questionnaires:** This strategy involves isolating the participant before they begin answering the survey or questionnaire to hopefully remove any social cues the researcher may present to the participant.

**Selection interviewers:** This strategy allows participants to select the person or persons who will be conducting the interview or presiding over the experiment. This method hopes that with a higher degree of rapport, subjects will be more likely to answer honestly.

The degree of effectiveness for each of these techniques or strategies differs depending on the situation and the question asked, and it is suggested to utilize a combination of these techniques to have the best chance at mitigating the effects of social desirability bias.

Therefore I have chosen the self-selecting sample method (Saunders et al., 2011) to gather qualitative data via semi-structured interviews. I have asked IT Managers in the organization randomly to cooperate with the case study and the respondents volunteering are most probably interested in the thesis topic and will have less bias.

Another bias can be caused by myself and can cause interviewer bias. To avoid occurrence I will ask questions objectively and with neutral tone and be self-aware on my non-verbal communication to not bias the results and increase the repeatability of my research. And I have used a clear start and stop script and invitation during the interview to create a comfortable setting for the interviewee (Myers & Newman, 2006).
The active search for secondary data for answering (parts of) the research questions has been considered as not beneficial, because I conducted an extensive literature review earlier in time. I have extended the interview questionnaire with a question if the participant has experience in the usage of Continuous Delivery Maturity Models (and has historical information to share) to avoid missing this information when it would be available. One of the CD Experts/Coaches shared after the interview a Continuous Delivery Maturity Model and self-assessment that I recognized as exactly similar to a self-assessment created by the organization DZone (and already found during literature study). Because of the commercial background of this assessment it is out of scope of my research. To search for secondary data outside my own organization is inconsistent with the proposed approach and is therefore not done.

The generalizability of my empirical research is limited because it is conducted in only one international organization. However the participants come from different countries (India, Netherlands, Romania and United Kingdom) and backgrounds. The fact that it is an embedded case study and the different participants are logically in other areas, have different backgrounds, experience, knowledge, business partners, other types of software (in-house development, COTS packages and/or outsourced software engineering) cannot justify that (all) my findings, conclusions and recommendations are generally valid. More research is needed in multiple organizations to increase and proof the broader adaptability and generalizability.

The completeness of information gathering can be improved by sharing information before the interview to prepare the interviewee and to already inform them with the interview questions. I will determine which information to share and in what format. It can also contribute to having a more efficient interview. I also have to avoid the sequencing bias and that participants bias other participants by guarding anonymity.

A critical remark on my empirical research is that almost all qualitative data is received by interviews and words being spoken, and no information being received by observation or follow-up. Observation as a technique is proposed to be included in further research.

I have invested and checked the access possibilities on different levels for the data gathering and the ethical aspects and feasibility. There are no significant particularities to report or act upon further.

### 5.6 Methods of analysis

All interviews are recorded. Immediately after the interview I draft a complete report of the interview (including all interview questions and answers) to prevent bias and produce reliable output data for analysis. Also this gives me the possibility to focus on data gathering (receiving complete answers on all of the research questions) and not be “distracted” by having to make notes constantly during the interview. And I have the possibility to replay parts of the interview if needed. The first interview had a duration of 49 minutes and it was a lot of work to transcribe and to analyze and extract the key information. The main reason was that I asked a lot of extra questions during the interview as a response of the answers of
the interviewee. In the subsequent interviews a much more structured approach was followed to keep them focused and short. I consequently focused on the script and the fixed structure of the pre-defined interview questions and that approach was more effective and efficient.

The type of information that I wanted to get out of the interviews can best be categorized as: “Opinion variables”. This is because I asked the own opinion of the interviewee compared to variables on behavior or characteristic variables.

Because the output of the interviews contains qualitative data, the main procedure to process and analyze the data requires the meaning to be summarized, categorized and structured. This should give enough foundation for the analysis to find similarities, differences and patterns and ultimately to draft conclusions in the upcoming chapters.

5.7 Questionnaire and interview results

The main research question is:

*What recommendations can be given to IT Managers for using a Continuous Delivery (CD) Maturity Model to implement CD in their organizations?*

Strengths and weaknesses of four broadly available Continuous Delivery Maturity models were analyzed and suggestions for their utilization were provided (Özcan-Top & Demirörs, 2013). This includes how sufficient the existing Continuous Delivery Maturity Models are in providing insight about an organization’s Continuous Delivery capabilities and what would be the next logical steps of maturing. To maintain focus and according to the Maturity Model design principles (Becker et al., 2009), I have chosen the most actual and most complete maturity model and that’s the Continuous Delivery Maturity Model by Andreas Rehn, Tobias Palmborg and Patrik Boström published on 6th of February 2013 for further empirical research (Rehn et al., 2013). Also I have asked Andreas Rehn permission to use his Continuous Delivery Maturity Model under the condition that I reference it accordingly. This was accepted and he showed great interest in my final thesis document.

The list of interview questions that was finalized after refinement and knowledge gained in expert discussion groups (Continuous Delivery Expert/Coach interviews) for the final empirical research approach is administrated in Appendix B. Insights from the literature study are already incorporated in that list of questions. And all these questions have the perspective of the (organization of the) IT Manager. In Appendix C the invite via email to the IT Managers can be found. I tried to make the invite neutral and avoid bias.

Hereafter the summary results of the 13 open questions are presented and consolidated into six sections. Section 5.7.1 starts with describing the importance of Continuous Delivery followed by Section 5.7.2 explaining the value of a Continuous Delivery Maturity Model and Section 5.7.3 contains the experiences of the interviewees with these kind of models. Section 5.7.4 presents which are the CD focus areas and Section 5.7.5 explains critics. Finally Section 5.7.6 summarizes all the recommendations extracted for the empirical research.
5.7.1 Importance of Continuous Delivery

Unanimously all interviewees stated that Continuous Delivery is very important. Being able to respond quickly to end-customer demand is something that requires much more proactive instant delivery capabilities from organizations comparing with more traditional waterfall approaches and software engineering methodologies. Market expectancy has become able to quickly adapt to different changes. Through the credit crisis and through IT-innovation in general: people’s expectations also changed quite a lot. The main reason is that we are moving into a much more digital and mobile-apps-enabled age where everybody expects to be able to go online and use their mobile device to find the answers they want. So it is all about understanding what users want and providing that.

Organizations should have an efficient framework to develop and put things live. It will have to be done in much shorter time frames, so the time between idea and hypothesis to real production and feedback should be shortened drastically. Quick time-to-market is nowadays of crucial importance.

Continuous Delivery is the key of that capability, also in complex environments with large numbers of teams, multiple technologies and solutions, and advocating to make smaller iterative steps. Continuous by definition means ‘all the time’, frequently, based on what your users are demanding. It will be super important and those organizations which operate according this new paradigm, will survive. Others, who are slower to react, will effectively have issues and potentially run out of business.

CD should improve the engineering capabilities and the quality of code of the bank or other organization. A lot of manual processes can be automated which are used for deployment or quality checking, and thereby you can deliver software better and faster. Continuous Delivery holds that promise of a big benefit of building quality in the delivery pipeline.

The feedback loop would be much more positive and the people will eventually have more time for, let’s say more fun development, than fixing bugs under pressure. CD makes also work more fun for people to move away from a lot of manual work and the boring work. Boring manual work is not a fit for high performance IT workforce.

5.7.2 Value of a Continuous Delivery Maturity Model

Many important values of a Continuous Delivery Maturity Model were listed during the interview rounds. By using a CD MM, organizations know where they stand, where to learn, where to improve, and know what can be shared to improve other parts of the organization.

This adds value and contributes towards better aligning and levelling the overall definitions and understanding of the CD areas through fact-based discussion. The model encourages clear communication, clear direction and clear steps. It assists in creating common understanding of where you can go and what would be the next stepping stones. It assists with defining incremental steps and a structured dialogue. It also helps by progressing these
smaller improvements and their business value can be explained to the businesses and contributes by building up credibility.

If you measure, you get understanding and facts. If you have that, you can decide and plan to improve. The main value is being able to understand, first of all, where you are on that scale of maturity. Which can be important to understand how you compare with peers in your sector and/or upcoming new competitors in your domain. But also to map out that route towards improved maturity and the key differences and benefits.

The added value of using a model could be also the transparency to the team and direct stakeholders and to assist in the analysis and discussion at where the most opportunity for the next step is.

So creating that CD roadmap based on planned actions which says: here is where I am now, here is where I need to be and here are the steps I am going to take to get there. That is the key value of the Continuous Delivery Maturity Model.

5.7.3 Experience with using Continuous Delivery Maturity Models

The three Continuous Delivery Experts/Coaches have significant experience for more than 5 years using Continuous Delivery Maturity Models in multiple organizations. All of them have all experience in using particular models, taking them out to clients and organizations and applying them. They are experienced in doing the assessment work, benchmarking where organizations are on those dimensions with those skills and providing advice on how they potentially could improve. The IT Managers have limited experience and showed interest in the Maturity Models, but have not actively searched for this instrument in the (recent) past. This is a gap that I have not investigated further, why are IT Managers not using this kind of tools already.

5.7.4 Continuous Delivery focus areas for Agile/SCRUM and DevOps

When you are looking at an organization, the most important internal assets are: people, process and technology. So there are potentially three key areas of focus. All of the seven interviewees (expect one) considered Culture and Organization as the most relevant Continuous Delivery area to focus on. Creating the right mindset to improve and encourage the DevOps teams as a group and the individuals in that group to make steps into the CD journey, is key. The interviewees stated that the culture, and how people are working with each other, is of most critical importance. The organizational topics can differ significantly between organizations: For example the organizational discrepancy of the quality of communications between teams can differ a lot in an organization. Then the focus area can be on teams communication and cooperation, resulting in actions to get the right people together, defining the improvement actions and have the situation structurally improved. The people dimension could be covered by the culture and organization category of Rehn’s CD MM. The process side of things is covered in terms of a number of the other tranches of Continuous Delivery Maturity Models. Automatic deployment and automated testing was
considered most important by one of the seven interviewees. Tooling and tooling
deficiencies are considered by all interviewees of secondary importance.

5.7.5 Critics on Continuous Delivery Maturity Models

There are seven main criticisms being mentioned in the interviews:

The CD Maturity Model is always somebody’s perception at a particular point in time. It is all
based on where an organization stands today. So the exact mapping in a model on what may
be good, better, excellent or whatever the metrics of experience or capability are, they are
based on somebody’s subjective interpretation.

Another concern present is around the language of the maturity levels. To use wording like
beginner, intermediate, advanced and expert has a bias risk. Those descriptions could be
considered as subjective and some people may not be comfortable with that and could
potentially negatively influence the adoption, usage and effectiveness of Continuous
Delivery Maturity Models. The risk that people (for example senior management) start first
of all with comparing teams and judging without knowing the context of a specific team,
without knowing where the team has come from and on what is have been achieved. And as
a result teams will mainly focus on creating a nicer dashboard score, instead of achieving the
real goals and associated benefits for the organization.

From experience, one of the interviewed CD Expert/Coach saw it being used as an internal
competition and management dashboard, reporting to qualify the current maturity of parts
of organization/department. A maturity model pretends everything is measurable, and most
probably that is not the truth.

Another critic is about the actuality. This depends on who further develops the Maturity
Model, and can be developed by many attributors. But they are based on a point in time,
which obviously then changes over time. So as the industry gets more mature in Continuous
Delivery, for example, you may be an advanced level today, but in a year or two that might
actually change to an intermediate level or you have made an intermediate to a beginner
level. Because of the advances made by the industry, peer group, etc. So Maturity Models
need to be revisited regularly to ensure that the dimensions stay accurate. And also you then
need to continue to find the model and find what the new expert level is. Because as the
expert level becomes the norm, which is again what happens, the contributors to the model
need to find out what the next higher level is. The main criticism to MM is that they are a
snapshot in time and it is difficult to continually find out the “what is pushing the
boundaries”. What is the leading edge capability?

In the interviews exact speed of delivery and time to market was not clear to the
interviewees. They all agree it should be faster, but no definition of clear thresholds can be
found in the CD MM. These thresholds are most probably not institutionalized and measured
in their organizations. How fast an idea prioritized on the backlog is released in production?
How long would it take your organization to deploy a change that involves just one single
line of code (Poppendieck & Poppendieck, 2003)? It is not always clear what a maturity level
means if you have knowledge or capabilities on a certain category or both. This is very
important to stimulate learning in organizations to transparently determine your “experts”
on certain levels and let them cooperate with the teams wanting to improve on these categories.

In many CD Maturity Models still the old “waterfall” definitions are used. This is not setting the right example and conflicts with the new way of work to always focus on added value for the end customer and avoid sub optimization.

CD Maturity Models are very focused on deployment of new software and functionalities, but also the growing importance of IT Operations, Reliability and Security should be completely covered in the Model coverage. This should also include monitoring and reduction of IT Operations and maintenance and IT Risk manual activities (for example environment status health check or environment reliability).

It is not clear where to start, even when you use a CD Maturity Model. And the level of abstraction can be improved or better explained. Some topics and improvement proposals require huge efforts, and other gains can be achieved with minimal efforts. It is not always clear how to deal with this. Sometimes CD MM seem complex and overwhelming and are not applicable for a specific organization.

How to achieve a simple, weighted scoring on the most important focus areas like culture and organization throughout the organization is not clear. There is a difference between how a team is assessed or how an organization consisting of a high number of teams assesses itself. It would be beneficial to be able to achieve aggregated results from the figures of specific teams. There are self-assessment tools available (for example from Praqma and DZone) for that purpose. This is a recommendation for further research and application.

The last criticism given by the interviewees is that there seems to be too much emphasis on the automated build and automated deployment, while there are other (or more) important improvements to achieve also.

5.7.6 Recommendations for Continuous Delivery Maturity Models

A common recommendation is to ensure that measuring of CD Maturity is done accurate and truthful. If the situation and as-is CD maturity is over-rated by the people involved, then important improvement actions will stay unidentified and not actioned as a result. And the organization will continue to not benefit from an increased CD maturity. IT Management should create overall adaption and correct usage of this tool throughout the enterprise.

Another recommendation is to use a CD Maturity Model to have teams create own CD roadmaps for improving in the different categories in time. So use CD MM for new insights and to structure discussions and arrange the model is approachable for the DevOps teams. Ensure the feedback loop is in place fully. So make a Continuous Delivery implementation and improvement plan and roadmap and perform very frequent Plan-Do-Check-Act with the team and relevant stakeholders. A shared ambition to achieve a pre-defined maturity at an agreed time could encourage the DevOps teams and give focus and clarity throughout the organization. With a CD Maturity Model progress against this target maturity could be measured and discussed.
This means to translate the model to an organization’s own language sometimes, because every organization has that. The people might not call things ‘deployment’, they might call them ‘installations’. The model should be tailored as it’s being applied to that particular organization. There will definitely be some pre-requisite research to understand the dynamics and alignment required before introducing and applying the model into an organization. The people should like and understand the model. And share this along teams and let them also invest in the improvements and increased maturity of other teams.

Planning and prioritization for Continuous Delivery improvements needs to come together into one clear governance and aligned result agreements for Business and IT. A forced ranked priority setting over the backlogs of the DevOps teams is very important. DevOps teams in our organizations have that much initiatives on their plate, so to ensure priority and focus for the important CD topic, clear planning and priority setting is a must have. IT Management should ensure alignment is reached with other (for example technical) developments, like migration of IT environments to new data centers, new tooling and enhanced security guidelines.

Create a group of enthusiastic CD practitioners in your organization and let them meet, share and show demonstrations. As early adaptors they can positively influence their environments, really exploit the advantages of CD and stimulate re-use of improvements in different parts of the organization.

Using production-like environments during testing is important. Otherwise you don’t test what will be the situation in production reality and that can be considered as disinvestment.

Do not setup a specific tooling team as it can easily become an “ivory tower” team. Then they start (only) making strategic, for example architectural tooling, decisions and create a gap with the DevOps teams. Standardization is very important to facilitate integrations in a chain of applications, re-use and lower tooling costs. The reasons and benefits of standardizing certain tools should be explained to and accepted by the complete organization. Continuous Delivery and Agile/SCRUM is about preventing and reducing hand-over moments. Setting up dedicated tooling and or automation teams into your organization is conflicting with this principle. And if there is a central team, they should not do the real work involved in releasing through the automated delivery pipeline. Advice is ok and beneficial, but not the real technical implementation, as you run a big risk of implementing a new dependency on this central team forever. No hand-over, no waste.

Start to align your management reporting throughout the different levels of organization. In many organizations the DevOps teams are being encouraged to a more mature way of information, monitoring and reporting and that conflicts and demotivates if older types of monthly reporting are still being requested and steered at by senior management. Empowerment of DevOps teams should result in a bottom-up reporting flow instead of top-down.

It is important that the model shows the interdependency between the areas. So doing zero time deployment requires, for example, culture and organizational changes. So it is
important that you consider that dimension as well into the overall cohesion. You cannot consider those dimensions in isolation. And it is important to have both the organizational as well as more technical areas being drafted in the CD MM. It is engaging and energizing the DevOps team members, and stimulates new ideas to be born in the process of planning for improvement.

Put emphasis on a uniform “definition of done” with all DevOps teams then automation can be enforced more and more teams will start using automatic tools.

Add examples and best case/use case practices to the model description. Add “value” as a component to the CD Maturity Model, because it is the most important selling point to the business to actually keep on prioritizing and implementing CD and it is also something that will come back, so it will be very nice instead of saying you only achieve the (business) benefits in the end.

Do not only use external consultants to come by and assess your organization (temporarily). It can be expected that this does not lead experts and IT Managers to deliver long-term structural progress.

Continue to use 5 levels, because it seems the appropriate number. For example having only 3 or 4 levels is less granular and considered as not detailed enough to have a clear distinction between maturity levels.
6 Results of analysis comparing theory and practice

Hereby an overview of the research results that is delivered as outcome of analysis and comparison of the literature review and the empirical research:

1. The relative importance of Continuous Delivery:
   • Continuous Delivery is very important for all organizations in their strive to a shorter time-to-market. This conclusion can be drawn after the literature review and the empirical research in the practice organization.

2. The added value of a Continuous Delivery Maturity Model for the practice organization.
   • In theory the main value of a Continuous Delivery Maturity Model lies in its use as an analysis and positioning tool. The framework and carefully developed set of criteria are invaluable to organizations with an urgent and persistent need to understand where they are in relation to the externally determined “best practices” of today.
   • In practice the added value assigned towards a Continuous Delivery Maturity Model lies more in its internal usage for alignment, discussion framework and for inspirational purposes to determine the next incremental steps.

3. Further prioritized overview of focus areas within the Continuous Delivery Maturity Model.
   • In theory no clear priority of focus areas could be found. The organization and its culture are probably the most important aspects to consider when aiming at creating a sustainable Continuous Delivery environment that takes advantage of all the resulting effects according to some authors. Others stated that strong team collaboration, tests automation, effective configuration management, deployment automation and good team culture are the major organizational focus areas advocated in CD to boost the effectiveness of a frequent delivery process.
   • In practice the organizational aspects are considered as most important “much haves” to be able to take the necessary (and often more technical) steps in the CD journey.

4. Criticism on the Continuous Delivery Maturity Model.
   • In theory only the following two criticisms were found: the draft maturity model is seldom validated systematically before publication and the intended quality and performance goals are not explicit in most cases.
   • In practice much more criticisms came to the surface, like objectivity issues, language description of the maturity levels, window dressing, actuality, measurement and threshold and definition issues, missing components, not clear where to start, complexity, weighted scoring and aggregation for enterprise, and too much emphasis on the automated build and automated deployment. This will lead to many recommendations in the next chapter.

5. Recommendations for IT Managers and for a future revised Continuous Delivery Maturity Model.
   • The main recommendations found in theory (without describing all “best practices” of Continuous Delivery) are to apply the Plan-Do-Check-cycle and
start to improve that part of the organization where people are open for changes and really willing to cooperate on improvements. They will have the best motivation to change. Value stream mapping can assist in further identifying these areas. Prioritize the improvements on their business case rationale by increasingly useful information and metrics. IT management plays a key role in determining gaps and adopting of the associated recommendations into their organizations.

- In practice I found that it is recommended to ensure that measuring is done honestly and reliably as a pre-requisite for adaption and usage throughout the enterprise. The use of a CD Maturity Model is to have teams create own CD roadmaps how to improve on the different categories in time. Translate the model to an organization’s own language. Planning and prioritization for Continuous Delivery needs to come together into one clear governance. Create groups of enthusiastic CD practitioners in your organization and stimulate re-use. Standardization is very important so standardize your tools and ensure continuous delivery is done with understanding and buy-in of the complete organization. Continuous Delivery and Agile/SCRUM is about preventing and reducing hand-over moments. Setting up dedicated tooling and or automation teams into your organization is conflicting with this principle. Start to align your management reporting throughout the different levels of the organization. It is important that the model shows the interdependency between the areas. The institutionalization of a uniform “definition of done” with all DevOps teams facilitates in setting a common foundation to work efficiently together in a application chain. Add examples and best case/use case practices to the model description. Add “value” as a component to the CD Maturity Model, because it is the most important selling point to the business.

6. Proposals for further research as part of the empirical circle.

- Although literature exists on organizational change, there seems to be very little, if any, research specifically focusing on introducing CD to an organization. Also I recommend investigating why IT managers are not actively reaching out to start using Continuous Delivery Maturity Models. Further research on this topic, e.g., understanding the challenges in more depth and developing strategies and practices to tackle them more effectively, will significantly help an organization’s smooth adoption of CD (Chen, 2014). And quality and performance goals are not directly stated in most cases. Further research on what organizations can achieve with the implementation of Continuous Delivery is desirable. Further research therefore could focus on establishing and explaining a relationship between the developments of an organization performance and the development of Continuous Delivery maturity.
7 Final conclusions and recommendations

Continuous Delivery is the new way of work within organizations being dependent on their software engineering and software management capabilities for shortening the time-to-market. And today the number of such organizations and the relative importance of Continuous Delivery to stay ahead in business is growing rapidly.

Continuous Delivery Maturity Models should be used as an (internal) analysis and positioning tool. But this is not enough and no guarantee for success. Use Continuous Delivery Maturity Models to draft specific improvement and transition plans. And the implementation of an iterative approach with frequent feedback loops according to a Plan-Do-Check-Act cycle is a necessity as well.

The organization and its culture are probably the most important aspects to consider when aiming at creating a sustainable Continuous Delivery environment. IT Management should create this context.

Strong team collaboration, test automation, effective configuration management, deployment automation and good team culture are the major organizational focus areas for implementing Continuous Delivery.

Many criticisms on (Continuous Delivery) Maturity Models exist and factual performance evidence is missing, also because of this rather innovative way of work and because adoption is still in an early stage.

However the results of organizations with high maturity on both the organizational and technical aspects of CD show that they are very successful today and outperform their peers.

IT Management should embrace this development and opportunity.

Planning and prioritization for Continuous Delivery needs to come together into one clear governance and the many benefits of the incremental improvements can start the fire that is needed for this paradigm change.
8 References

Academic Peer-Reviewed References


**Other Secondary Sources**


## Appendix A: Similarities and differences between the four CD Maturity Models:

### Table 2: Similarities between the four Continuous Delivery Maturity Models (by similar analysis as Kohlegger et al. in 2009)

<table>
<thead>
<tr>
<th>Model Name/Category</th>
<th>Criteria</th>
<th>CD MM 1 Humble</th>
<th>CD MM 2 Benefield</th>
<th>CD MM 3 Minick</th>
<th>CD MM 4 Rehn</th>
<th>Frequency</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. How do elements change in time?</td>
<td>change in number</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>change in nature</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>4</td>
</tr>
<tr>
<td>2. Direction of change</td>
<td>Increasing change</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>4</td>
</tr>
<tr>
<td></td>
<td>Decreasing change</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3. Maturing object</td>
<td>person</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>document</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>infrastructure</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>4</td>
</tr>
<tr>
<td></td>
<td>product</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>service</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>group</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>4</td>
</tr>
<tr>
<td></td>
<td>team</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>4</td>
</tr>
<tr>
<td></td>
<td>community</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>4</td>
</tr>
<tr>
<td></td>
<td>process</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>4</td>
</tr>
<tr>
<td></td>
<td>routine</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>4</td>
</tr>
<tr>
<td></td>
<td>structure</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4. Model design</td>
<td>Iterative (one path)</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>4</td>
</tr>
<tr>
<td></td>
<td>Cyclical (many turns)</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5. How do levels built on other levels?</td>
<td>Upper level comprises lower level</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>4</td>
</tr>
<tr>
<td></td>
<td>Upper level is new concept</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6. How to come from one Level to another?</td>
<td>Defined goals have to be fulfilled</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>4</td>
</tr>
<tr>
<td></td>
<td>Matures implicit</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7. Degree of detail</td>
<td>One trigger per stage</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Many triggers per stage</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>4</td>
</tr>
<tr>
<td></td>
<td>No triggers per stage</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>8. Method of goal benchmarking</td>
<td>Metric based</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Non-metric based</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>4</td>
</tr>
<tr>
<td>9. Certification availability</td>
<td>Certification is available</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>No certification available</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>4</td>
</tr>
<tr>
<td>10. What means maturing?</td>
<td>Change in quality</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>4</td>
</tr>
<tr>
<td></td>
<td>Change in capability</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>4</td>
</tr>
<tr>
<td></td>
<td>Change in risk</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Other change</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>11. Number of stages</td>
<td>Metric value</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>20</td>
</tr>
<tr>
<td>12. Description of stages</td>
<td>First stage</td>
<td>Regressive</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Second stage</td>
<td>Repeatable</td>
<td>Level 1</td>
<td>Base</td>
<td>Base</td>
<td>4</td>
</tr>
<tr>
<td></td>
<td>Third stage</td>
<td>Consistent</td>
<td>Level 2</td>
<td>Beginner</td>
<td>Beginner</td>
<td>4</td>
</tr>
<tr>
<td></td>
<td>Fourth stage</td>
<td>Quantitatively managed</td>
<td>Level 3</td>
<td>Intermediate</td>
<td>Intermediate</td>
<td>4</td>
</tr>
<tr>
<td></td>
<td>Fifth stage</td>
<td>Optimizing</td>
<td>Level 4</td>
<td>Advanced</td>
<td>Advanced</td>
<td>4</td>
</tr>
<tr>
<td>13. What do level descriptions include?</td>
<td>Trigger descriptions</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>4</td>
</tr>
<tr>
<td></td>
<td>Activity descriptions (tasks, processes)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Conceptual level description</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>14. Level skipping</td>
<td>Explicitly allowed</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Not recommended</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>4</td>
</tr>
<tr>
<td>15. Are there parallel maturing processes possible for one unit?</td>
<td>Parallel maturing is possible</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>4</td>
</tr>
<tr>
<td></td>
<td>Parallel maturing is not possible</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>16. Where do assessment data come from?</td>
<td>Interviews</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Documents</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Questionnaire</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>4</td>
</tr>
<tr>
<td></td>
<td>Data</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>17. Tool support</td>
<td>Supported by assessment model</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Supported by software tool</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Not supported by tool</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>4</td>
</tr>
</tbody>
</table>

Table 2: Similarities between the four Continuous Delivery Maturity Models (by similar analysis as Kohlegger et al. in 2009).
<table>
<thead>
<tr>
<th>Model Name/ Category</th>
<th>Criteria</th>
<th>CD MM 1 Humble</th>
<th>CD MM 2 Benefield</th>
<th>CD MM 3 Minick</th>
<th>CD MM 4 Rehn</th>
<th>Frequency</th>
</tr>
</thead>
<tbody>
<tr>
<td>18. Existing Model usage</td>
<td>Built on existing model</td>
<td>1</td>
<td>1</td>
<td>2</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Not built on existing model</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>19. Is there a “not started” stage?</td>
<td>there is a “not started” stage</td>
<td>1</td>
<td></td>
<td></td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>there is no “not started” stage</td>
<td></td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>20. Number of Goal Levels</td>
<td>metric values</td>
<td>6</td>
<td>7</td>
<td>4</td>
<td>5</td>
<td>22</td>
</tr>
</tbody>
</table>

| Category 1 | Build management | Automated Builds and Configuration Management best practices | Building | Build & Deploy | |
| Category 2 | Environments and deployment | Automated Deployment and Backout | Deploying | Design & Architecture | |
| Category 3 | Release management and compliance | Interlocked Delivery and Interface Integration testing | | | |
| Category 4 | Testing | Automated Regression Testing | Testing | Test & Verification | |
| Category 5 | Data management | Code Quality Metrics | Reporting | Information & Reporting | |
| Category 6 | Configuration management | Performance and Scalability Testing | | | |
| Category 7 | Test Driven Development (TDD) | | Culture & Organization | | |

Table 3: Differences between the four Continuous Delivery Maturity Models (by similar analysis as Kohlegger et all in 2009).
Appendix B: Questionnaire for semi-structured interview:

*All these questions have the perspective of the (organization of the) IT Manager.

**Generic Continuous Delivery and Maturity Model questions:**

1. **How important is Continuous Delivery for your organization and why?**
   - Expected feedback is that Continuous Delivery and maturing is very important for the organization. Specific reasons could include: improved time-to-market, lower costs/higher efficiency, keep up with competitors etc.

2. **What is your experience in the usage of Continuous Delivery Maturity Models?**
   - Theory is that CD Maturity Models are new and I expect only Humble’s MM is known by coaches and IT Managers. Overall I have not seen the usage of CD MM’s in the organization (except one example within BackOffice IT -> there the Rehn CD MM is being used at least for one application).

3. **What should be the main added value of (using) a CD Maturity Model?**
   - The main value of a maturity model lies in its use as an analysis and positioning tool. This to give structure and understanding to the implementation of Continuous delivery and its core components. Maturity models will give you a starting point and a base for planning the transformation of the company towards Continuous Delivery (both descriptive and prescriptive value).

4. **What are the CD focus areas most relevant for your organizational scope by working Agile/SCRUM in DevOps teams and why?**
   - The organization and it’s culture are probably the most important aspects to consider when aiming to create a sustainable Continuous Delivery environment that takes advantage of all the resulting effects.

5. **What critics do you have on CD Maturity Models and why?**
   - Usage, simplification, mapping, unclarity, unclear explanation, no performance evidence.

**Specific questions on the Continuous Delivery Maturity Model (CD MM 4) of Rehn:**

1. **How is this MM helpful to determine the status of CD maturity in your organization and why?**
   - Maturity Level of own organization can easily be mapped in the MM.

2. **How is this MM helpful to determine the next steps of implementing CD in your organization and why?**
   - Description of next level MM is clear and will support the planning and improvements cycle.

3. **What should be extended in the MM and why?**
   - According to theory this CD MM should be complete. But it could be that more categories/focus areas/other items are being mentioned. Extensions can also consist of extra recommendations, for example: Re-use existing Continuous Delivery Maturity Models and thereby making the as-is assessment more easy to map and the next step easily understandable as an improvement.

4. **What recommendations can be given next to the MM to implement CD efficiently and effectively in your organization and why?**
Exploratory question, could lead to many proposals. Out of the scientific theory I found (amongst others):

- CD MM should become integral part of Plan-Do-Check-Act cycle.
- Start with CD where the most challenges are in the organization and people are suffering.

5. How will you use this maturity model for self-assessment in the future and why (not)?
   - In what frequency and with who (engineers, other IT managers, business partners etc).

6. Why would you recommend this MM to other IT Managers working inside or outside your organization?
   - Because of the value and meets model quality criteria (fitness for purpose, completeness, definition of agile levels, objectivity, correctness and consistency).

7. What have you learned from practicing with the CD Maturity model an why?
   - Open questions, many answers possible.

8. What can be improved in the MM and why?
   - Related to and partly repeating of question 3 and 4. And good check question and final question to receive relevant information from the interviewee.
Appendix C: Invite example for semi-structured interview:

From: Wezelman, W. (Wouter)
Sent: maandag 25 mei 2015 15:50
To:
Subject: Interview on Continuous Delivery and Maturity Models.

Dear Colleague,

Can I please ask you for taking the time (max 30-45 minutes) for an interview for the practical research as part of my thesis on Continuous Delivery and Maturity Models?

*Be informed that the Continuous Delivery maturity of your organization and/or department is NOT part of my research and this interview.

I would like to ask you in total 13 open questions according the semi-structured interview approach. All these questions have the perspective of the (organization of the) IT Manager.

You don’t need to prepare for the interview (as a small preparation it could be beneficial for your own time, if you can already go through the below 13 questions).

Also I will bring the below explanation and chart of the Continuous Delivery Maturity Model of Rehn (see below). Feel free to already have a look, otherwise I will quickly explain it during the interview. This will take a only a few minutes.

The interview is in English and will be recorded on my phone and immediately deleted after filing the results within a few days. The recording itself will not be used in any other way, as discretion is key. I use recording technique to shorten the interview duration, hence I do not need to make notes during the interview.

If you have any question or remarks, please let me know. I will also ask this before and after the interview.

Below you can find the 13 questions, and the explanation of the Continuous Delivery Maturity Model of Rehn (including model chart).

Please let me know if I can ask your cooperation? It’s very much appreciated!

Greetings Wouter Wezelman
9 Interviews

9.1 Interview 1: CD Expert & Coach

<table>
<thead>
<tr>
<th>Interview 1</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of minutes:</td>
</tr>
<tr>
<td>Number of speakers:</td>
</tr>
<tr>
<td>Language:</td>
</tr>
</tbody>
</table>

Sp. 1: Hoe belangrijk is Continuous Delivery voor je organisatie en waarom?
Sp. 2: Heel belangrijk. Volgens mij waarom wij binnen onze organisatie Continuous Delivery doen is… Ik persoonlijk denk dat het zo’n ingewikkeld landschap is, dat als je daar grote tanker bewegingen in wil maken, met waterval methodes, dat je meer schade aanricht dan dat je kan wijzigen. En op het moment dat je met kleine stapjes software development doet, kleine en korte iteraties, het veel makkelijker af te stemmen is met de omgeving. De reactiesnelheid is gewoon veel groter. Ondanks dat je een heel complex en groot landschap hebt, je reactiesnelheid op bijvoorbeeld met wat er in de markt gebeurt, dat die gewoon veel sneller is dan dat je met waterval kunt bereiken denk ik.

Sp. 1: En wat zou jouw threshold van gewenste snelheid zijn? Hoe snel moet iets live?
Sp. 2: Ik denk dat binnen de IT wereld, zoals we dat nu bij de banken kennen, de reactiesnelheid van banken of marktontwikkelingen veel sneller moeten. Want als je kijkt naar banken zelf… Voor rentetarieven gaan we de klanten niet meer weghalen, want dat is het hele verhaal, volgens mij. Dus het gaat alleen maar over snelheid en over service. Nou, waar wordt je op dit moment op gewaardeerd? Op je externe factoren. Vroeger was dat rentetarieven en nu is het je mobiele apps en hoe snel je daarmee bent met vernieuwing? Het is bijna een auto geworden, qua bedrijf.

Sp. 1: Ja. Dus daarin verandert echt de markt.

Sp. 1: En als we dat niet zouden doen? Wat zou er dan gebeuren met de organisatie?
Sp. 2: Ik denk dat we dan ooit een keer out of business raken.

Sp. 1: Dus dan verliezen we klanten.
Sp. 2: Absoluut.

Sp. 2: Ja, dit is mijn ervaring. Weet je, ik heb natuurlijk bij mijn vorige organisatie ook met een Maturity Model gewerkt en dat hebben we toen zelf in mekaar gezet. En dan kom ik terug op wat mijn ervaring ermei is. Ik vind het doodeng, die maturity modellen. En waarom? Omdat het een model is dat suggereert dat het meetbaar is, maar dat is het niet. Het is een beleving. Hoe mature ben je nou? Meet je dat nou van het aantal software deployments per
week? Als je dat maturity model ziet dat wij toen hebben gebruikt, daar werd er door managers ingevuld wat het maturity niveau was van hun team op basis van softe meetlat waar langs gemeten werd. Zoals ik het ervaren heb, maar dat is een negatieve ervaring, is het dat we zo mature zijn dat we de experts niet meer nodig hebben. Maar dat is heel fictief ingevuld. En los van het feit of dat waar is, vind ik dat het gevaar van zo'n model. Als je geen referentie hebt, dat iedereen 'm invult naar z'n eigen referentie. En binnen die organisatie zoals zijn er 60 managers geweest die een soort competitie met elkaar aan gingen, want continuous delivery en agile is ook een soort klein competitiegevalletje. Daar kunnen ze dingen beter en sneller en dat is het competitie-effect dat er in Continuous Delivery zit. En dat zie je ook in management komen, dan ga je dit soort rapportages competitief invullen. Dat vind ik soms een beetje eng.

Sp. 1: Dus dan wordt het meer een model waarbij managers elkaar een beetje de loef afsteken, om te laten zien hoe goed en ver hun afdeling is.
Sp. 2: Ja, dan wordt het een soort dashboardrapportage en die vind ik dodelijk als je de verhaal erachter niet kent.

Sp. 1: Oké. Dat snap ik. Heb je zelf weleens vanuit het internet of in een artikel een Continuous Delivery Model gezien?

Sp. 1: Dat is echt een voorwaarde.
Sp. 2: Dat is echt een voorwaarde. Hoe eerlijk durf je te zijn?

Sp. 1: We gaan nu naar vraag 3. Wat zou de toegevoegde waarde zijn van het gebruik van een Maturity Model? Dus onder de voorwaarde dat je het werkelijk gebruikt voor je eigen organisatie, in plaats van anderen de loef af te steken.
ik een ander verhaal van Continuous Delivery. Iedereen focust zich heel erg op z’n eigen backlog, op z’n eigen eiland, op z’n eigen systeem en op z’n eigen omgeving. En wat je nu vraagt is om over de grens te gaan kijken naar hoe een ander het doet. En mensen nemen daar de tijd niet voor. Dat vind ik echt heel jammer. En dan vind ik dat de added value van een Maturity Model daaruit voort komt. Als je dat kan, dan haal je alles eruit wat je nodig hebt om te verbeteren.

Sp. 1: Wat bedoel je precies?

Sp. 1: Oké. En je noemde net een voorbeeld dat je dat dan wel nog laat doen door dezelfde collega’s in de organisatie, dat is wel nog goed genoeg? Of moet het volgens jou helemaal door buiten?

Sp. 1: Dus dat zou je niet gaan doen. Externe consultants die partij…
Sp. 2: Je mag best begeleiding vragen, maar je moet ze niet laten beoordelen. Want een extern bedrijf heeft ook een andere opdracht. Iedereen reageert ook op die opdracht en niet meer op z’n eigen maturity. Dat vind ik heel gevaarlijk. Zou ik niet doen.

Sp. 1: Wat zou er nog meer een toegevoegde waarde kunnen zijn van Continuous Delivery Maturity Model en het gebruik ervan?

Sp. 1: Oké, dat is een leuke.

Sp. 2: Ja. Al is het alleen maar door te vertellen dat iemand er al iets mee gedaan heeft. Dat “iets” is dan te achterhalen en dat wordt het lastige, dat is het moeilijke van die meetlat van het Maturity Model. Wat is dan “iets” en hoe waardeer je dat dan om te zorgen dat dat
Maturity heeft? Dat is het moeilijkste van het model. Iemand heeft iets gedaan en hoe mature is hij daar dan mee? Want als hij als eerste begonnen is, is hij het meest mature. Of is hij als laatste begonnen en heeft hij het binnen een week gedaan, is hij dan meer mature dan iemand die er drie maanden over gedaan heeft, vier maanden geleden? Dat vind ik wel een ding. Dat vind ik het gevaar van het Maturity Model. Aan de andere kant is DevOps ook gewoon re-use; hergebruiken van een ander. Je bent gek als je iets nieuws aan het verzinnen bent, terwijl een ander het al bedacht heeft. Het ging toch over snelheid? En dat laat je hopelijk ook zien door dit soort dingen te delen in het Maturity Model. Je moet het niet te theoretisch maken. Je moet het wel een beetje praktisch maken, anders hebben teams er niks aan. Dat is management gepraat.

Sp. 1: En dat is mijn volgende vraag; wat zijn voor jou in een Continuous Delivery de belangrijkste area's om op te focussen? Want er zijn er verschillende hier. Het zijn er vijf. Information and Reporting, Test and Verification, Build and Deploy, Design and Architect en Culture and Organisation. Maar het kan er één van deze zijn, het kan er ook één hele andere zijn.

Sp. 2: Ik vind het moeilijk om te zeggen dat ik één van deze vijf het belangrijkste vind. Of een zesde die ik zelf ga verzinnen. Wat ik een heel belangrijk focuspunt vind is dat je de nadruk legt op de teams. En als je de teams eenmaal in de loop hebt, dus een Maturity Model van de teams, als je die en die loops hebt, ga je zelf excelleren waar je eng van wordt. Als je nu ziet dat we nog steeds een stukje management driven zijn, dan doen ze het niet voor zichzelf.

Sp. 1: Je bedoelt met teams echt dat je het zo klein mogelijk moet maken?

Sp. 2: Niet alleen zo klein mogelijk. Wat ik bedoel met teams is dat ik het cultureel aspect van dit lijstje vind ik dus… Als je zegt over cultuur: “we doen het allemaal anders”. Nee, als het vanuit jezelf komt, gaat het vanzelf. En de teams moeten het uit zichzelf zien te krijgen. Er is een discussie over reliability, dat vind ik het mooiste voorbeeld. Er gaan mensen naar die training toe en wordt er gevraagd “waarom zit je hier nou?” “Nou, ik moest hier naartoe van m’n manager”. Het gaat over configuratie management, het meest saaie gebied van IT. En dan denk ik: je zit hier niet voor je manager. Je zit hier, omdat je wil verbeteren. En als jij als developer in zo’n training configuratie management, wat je ………vind, daar zit met de gedachte “als ik dit goed doe, kan ik m’n hele Continuous Delivery een boost geven, omdat ik geen tikfouten meer maak en ik heb het geautomatiseerd, want daarom zit ik in deze wereld”, dan komt het uit jezelf en dan wordt het ook leuk. Maar iemand die zegt: “ik doe het nog drie keer, ik heb geen zin om het voor de vierde keer te doen” en niet die mindset maakt van: “als ik dit voor de vierde keer ga doen, ga ik het automatiseren”. Als je dat stapje maakt, dan maakt het mij niet uit of je dat met Continuous Delivery doet, met een Python script of met een tool. Dat boeit dan niet. Als je die slag maakt, dan word je een winnaar. Daar ben ik van overtuigd.

Sp. 1: Dat is echt op individu niveau, toch?

Sp. 2: Ja, maar ook op team niveau. Maar het begint absoluut bij het individu.

Sp. 1: Maar dan heb je een paar van die individu’s nodig.

Sp. 2: Ja, je hebt dat soort rakkers nodig die die drive hebben. Dat heb ik hier ook. Ik heb in mijn team een bulldozer zitten, en hij automatiseret ook. Hij gaat soms heel ver want hij is nu complete UNIX instellingen aan het veranderen en zegt: “op die manier gaan we het terugbrengen naar de infra teams. Ik zeg: “denk je dat zij dat accepteren?” Ga ze het maar

Sp. 1: De volgende vraag is: welke kritiek heb je op Continuous Delivery Maturity Modellen?
Sp. 2: Ik heb er al een paar genoemd.

Sp. 1: In mijn woorden zou dat vooral zijn dat je het niet moet gebruiken om elkaar de loeven af te steken of dat het alleen een soort managementtool is. Welke kritiek heb je nog meer?
Sp. 2: Nou, zorg dat ze pragmatisch zijn. Maak het niet te veel een rapportagetool. Als je ook echt wil dat er wat mee gebeurt, wat erin staat, moet je het pragmatisch houden. En dan moet het dus heel aansprekend zijn voor teams. Net zoals smell-o-meter tussen Fontaine Bleu, ik heb iedereen moeten uitleggen wat Fontaine Bleu was. Maar dat verschilt… iedereen wist was Fontaine Bleu was, maar geen idee waar Fontaine Bleu lag…. Dus als je dat noemt, denk je “het zal wel”. Dus je moet het wel voor de mensen heel erg aanspreekbaar maken, want zij moeten er wat mee gaan doen en niet de managers. Dus je moet echt zorgen dat zij er wat mee gaan doen, dat is de kracht van Maturity. Mijn kritiek is dat ik het soms te veel managementdefinities vind hebben.

Sp. 1: Een beetje een suggestieve vraag, maar dus niet de managers moeten met een Continuous Delivery Model aan de gang gaan, maar ook de medewerkers zelf?

Sp. 1: Oké.
Sp. 2: Hm… heb ik nog meer kritiek? Ja, als je met kleurtjes gaat werken, wordt het een stoplicht. Dat vind ik eng. En dat zie je wel vaker, dat er echt kleurtjes gebruikt worden. Zoals groen, wat is groen nou? Wat staat er nou? Groen is doorrijden, maar ik kijk toch altijd even naar links en naar rechts.

Sp. 1: Ik heb verschillende Continuous Delivery Modellen bekeken, dit is een model van Rehn en er zit ook een soort beschrijving bij. Er zijn vijf levels en vijf categorieën. Elk level heeft een bepaald onderwerp en vervolgens is er een soort base, dus wat elke organisatie is als je nog niet echt verbeterd hebt daarin. En vervolgens heb je dan een aantal stappen met een aantal criteria waaraan volstaan kan worden en dan zou je dus die hogere volwassenheidsniveau hebben. Ik zou daar graag een aantal vragen over stellen. Ik begrijp dat je dit niet al jaren hebt toegepast, maar dat maakt ook niet uit.
Sp. 2: Weet je, de base, beginner, intermediate, adanced, experts zijn gewoon vijf niveaus
die ik ook wel herken uit modellen die we bij mijn vorige organisatie hebben gebruikt. Dus het feit dat je vijf levels definieert. Wat ik een beetje mis in je uitleg over de experten en de vijf stadia waarin je kan verkeren is dat als je een expert bent op bijvoorbeeld het gebied van Design and Architecture, wat ben je dan als expert? Ben je expert Design and Architect of ben je een voorbeeld voor de ander? Kom ik weer terug op het leereffect. Dat mis ik hier een beetje bij dat het een meetlat is. Wat doe je nou naar een ander toe? Dat zou ik 'm dan wel weer in het culturele aspect, denk ik. Culture and Organisation. Ik denk wel dat je hier wel kan pinpointen van: “joh, ik ben goed in Build en Deplo, maar nog lang niet in Verification.” Die maturity moet je willen… dat zit ‘m dan wel weer hier in het culturele aspect, denk ik. Culture and Organisation. Ik denk wel dat je hier wel kan pinpointen van: “joh, ik ben goed in Build en Deplo, maar nog lang niet in Verification.” Ik geloof wel dat je nu kijkt naar hoe wij het binnen ons toepassen, denk ik dat we het ook op die manier benaderen om te focussen op waar de verbeteringen in zitten. Dus daar zou helpen. Alleen de logica van DevOps… Een voorbeeld is als ik kijk naar wat bijvoorbeeld nu bij end-to-end testen in onze organisatie wordt gedaan, wie is er bezig om te zorgen dat, automatisch testing moet je doen, maar is het logisch om daarmee te beginnen? Of was het veel logischer om overal Continuous Development in te leveren? Dus overal Nolio flows implementeren. Dat je weet dat je omgeving precies gelijk is. Want je kunt zoveel testen wat je wil, maar als je omgeving niet gelijk is, weet je het verhaal. Punt. Dus is het logisch om op die manier te beginnen? Die logica in maturity haal ik er hier nog niet uit. Gaat het me hierbij helpen? Dat denk ik wel, maar ik moet zelf de logica verzinnen.

Sp. 1: Ja, die mist inderdaad. Hoe Rehn het uitlegt in dit model is dat je op alle verschillende lagen zou kunnen zijn. Er kan één of twee verschillen zitten tussen een bepaalde maturity niveau, maar eigenlijk idealiter zou je op al die lagen moeten proberen zo’n organisatie steeds tegelijkertijd een stapje te laten maken. Maar zoals jij het ook aangeeft, dat kan
natuurlijk wel uitkomen als, aardig veel. Want dan ben je meteen op vijf verschillende onderdelen een stap aan het maken.

Sp. 2: Als je nu kijkt naar E2E testing het staat hier niet in, is een heel goed doel. Maar wat moet je doen om E2E testing goed te doen? Je moet gaan automatiseren, je moet dus je omgevingen gelijk gaan trekken, dus je moet Continuous Delivery ingevuld hebben. Dan moet je het kunnen monitoren. Je moet alles tegelijkertijd gaan doen, wil je het goed laten slagen. Dus is het logisch om met E2E testen te beginnen of is het logischer om ergens anders te beginnen? Dat is het moeilijke van dit soort dingen. Want ook dit moet je in kleine stapjes tegelijk doen, dus je moet je ambitie ook niet te hoog neerzetten, vind ik.

Sp. 1: Kan dit model helpen bij het bepalen van die volgende stap?
Sp. 2: Ja.

Sp. 1: En waarom vind je dat?
Sp. 2: Nou, zoals ik het net even gescand heb, denk ik dat het wel beschrijft wat…. Als je kijkt naar de levels per categorie, dat je daarin wel je ambities kan beschrijven. En vanuit de beschrijving die ze hierin hebben gebruikt, moet je in staat kunnen zijn om je volgende stap te kunnen definiëren. Dat vind ik wel. Je haalt hier heel makkelijk uit wat het verschil is tussen een base en een beginner, hetzelfde is met beoordelen of medewerkers. Hoe ga je nou van drie naar vier qua beoordeling? Dan moet je een aantal dingen laten zien die kun je beschrijven. Die kun je zelf bedenken. Nou, dat kan je hier ook. Gevaar wat erin zit is dat als je hier met zeventig teams tegelijkertijd dit model gaat, dat die zeventig interpretaties plaatsvinden tussen base en beginner. Dat is het risico. Dus daar moet je misschien wat afspraken over maken, want het zou toch jammer zijn als je zegt dat je een expert bent op testing en er komt iemand van een ander team langs die zegt dat je je script niet hebt gecontroleerd. Dus daar kan het je zeker bij helpen. Ik vind sowieso dat je het niet te veel moet theoretiseren; een model is een houvast. En we tanken hier allemaal benzine, alleen de één gebruikt een andere motor en verbruikt daardoor meer, maar de standaard is gewoon die benzine. En hoe je ’m gaat toepassen, bepaal je zelf. Voor mijn gevoel kan je het hier best wel meetbaar maken en beschrijven wat je moet doen om van base naar beginner te gaan. Dat kan. Maar dat kan met elk model, ook met die andere modellen. Maar de voorwaarde is weer: wees daar eerlijk in. Wees daar objectief in. Dat moet je echt doen, anders werkt het niet. En dat is het gevaar ook weer. Je kunt jezelf voor de gek houden. Dat is niet zo moeilijk.

Sp. 2: Ja, als je niet uitkijkt begin je als eerst met je design en architectenstuk te optimaliseren. Dan is het wachten tot jij een keer klaar bent. Dat moet je niet doen. Maar het is ook niet handig om te zeggen: “team, verzin maar wat!” Een voorbeeld van XFB en SFTP. Ja, vervelend. De teams hadden bedacht dat het SFTP ging worden en toch hebben ze bepaald dat het XFB wordt. Ik vind daar wat van. Maar dan is de logica om het andersom te doen, “just do it”, maar het blijft wel toetsen. En nu is iedereen pissig over het feit dat het …. XFB geworden is. Dus killing als je het hebt over een lerende organisatie, management driven. Dus er is wel een rode draad te trekken. Ik weet niet of die erin zit, heb ik ’m niet voor gelezen, maar er moet een rode draad zijn in de volgorde van deze stappen. In de logica van deze stappen. Het is helemaal niet erg om bijvoorbeeld als eerst met mijn file transfers praten om dat ingericht te krijgen. Maar begin dan wel met je architectuur te praten over
welke kans je maakt. Hetzelfde als wat ik nu doe met IT voor IT heb. Iedereen kijkt naar mij
welke test tools zijn nu standaard. Die zijn er niet. “Wat zou je me adviseren?” “Ik denk dat
dit de rode draad wordt”. Schiet me niet dood als het dat toch niet wordt, maar je mag van
mij wel verwachten dat ik een redelijke lijn kan trekken in de rode draad. Nou heb ik niet een
functioneel testje van applicatie x. Nee, het feit dat ‘ie op een tool met de api kan praten, dat
vind ik veel belangrijker. Dat is een beetje het spel tussen een engineer, die wil testen en de
meer strategische architectuur. Die kunnen tegelijkertijd beginnen. Niet los van mekaar. Dat
is m’n grootste zorg bij DevOps, iedereen blijft op z’n eiland zitten en gaat zelf bepalen wat
voor hem het beste is. Dat kun je met geen één model tegenhouden.

Sp. 1: Ik denk dat je moet aangrijpen dat dit niet genoeg is om de benodigde stappen in je
transitie te maken, maar dat je op basis van bijvoorbeeld deze categorie aan de gang gaat
met een team die een apart implementatieplan maakt waarbij je net aangeeft dat ze dingen
gaan doen op basis van X of B en dingen gaan met Design and Architecture. Ik denk dat dat
een goede aanvulling is. Welke aanbevelingen kun je in de breedte geven aan organisaties
om naast het Maturity Model Continuous Delivery te implementeren? Ik heb al een aantal
goede tips van je gehoord. Heb je nog andere tips? Je vond die cultuur heel erg belangrijk, je
moet dingen klein maken, je gaf net nog een hele belangrijke aan: blijf steeds toetsen of je
on track ben, maar ook of er zaken veranderen.

Sp. 2: Ja, ook dat doe je agile. Als je vier maanden niet toetst en je komt er in één keer
achter dat je het bent vergeten, heb je vier maanden verspild. Dus ook dat moet je continu

Sp. 1: Één van de stukken die Rehn aangeeft is dat hij ook zegt: maak in je organisatie een
apart team. Richt die in om de automatisering te doen. Jij hebt zo’n apart team, maar
daarnaast hoor ik je ook heel erg zeggen dat het team zelf meters moeten maken.

Sp. 2: Ik ben het ook niet met hem eens.

Sp. 1: Dus wat is het idee?

Sp. 2: Het is geen zwart-wit discussie. Wat je nu ziet bijvoorbeeld bij een nieuw initiatief is
dat zij nu aan het bepalen zijn wat er gebruikt moet worden aan tooling. Nou, wat je ziet
gebeuren is dat teams denken: “welke randebiel heeft…” Iedereen gebruikt GitHub en zij
bepalen Gitlab. Hebben ze een goede reden voor, maar zij bepalen. Dus ze zijn een ivoren
toren geworden. Dus je gaat toch weer wat opleggen, wat ik op zich begrijp. Je kunt
honderdduizend tools met elkaar gaan koppelen, maar dat werkt niet. Dus wat ik vind in de
taal van: “ja, je moet wel een centraal team hebben”; dat wat mijn team moet doen en doet is
ten eerste, doe niet het werk voor een ander. Want ik ga dan weer wat opleveren, dus ik ga
een overdrachtsmoment creëren. Waarom doen wij Continuous Delivery? Zodat we geen
overdrachtsmomenten meer hebben. Dus daar ben ik tegen. Dus ik wil geen
overdrachtsmomenten hebben. Dus mijn teams maken niet dingen die anderen moeten gaan
implementeren, want ik weet niet of dat werkt. En dan krijg ik van: “ja, maar bij mij werk het
niet, want dan heb ik een overdrachtsmoment”, en dat wil ik niet. Dus ze gaan samen met
die teams dingen implementeren. Dus ik doe niks voor ze. Ik geef ze een handleiding en hier
zeg ik Nolio flow. Dat is heel simpel. Ik kan veertig keer een team vragen of veertig keer een
nolio flow van maken of ik kan één keer een plan inleveren en zeggen: “van nolio is dit de
standaard sowieso, daarnaast mag je je aanpassingen op je applicatie verwachten”. Maar
dat moet je wel doen. In die zin ben ik het wel met hem eens, maar het moet niet zo zijn dat
ik ga zeggen wie wat precies moet doen. Dan krijg je een infra cultuur. Je weet hoe dat op dit
moment werkt; zij begrijpen geen zak van wat wij hier aan het doen zijn en wij begrijpen niets

Sp. 1: En over gebruik, hoe zou je dit Continuous Delivery model of zelfassessment gebruiken in de toekomst of waarom niet?
Sp. 2: Of ik het zelf zou gebruiken?


Sp. 1: Dus eerst zelf invullen en dan is het belangrijkste het samen bespreken.
Sp. 2: Ja!


Sp. 1: Dus dat wil je ook echt? Dat toetsen?
Sp. 2: Dat wil je ook echt. Dat wil je toetsen. Dus er is een groot verschil tussen een beoordelaar en een expert. Daar zit echt een groot verschil in. Je kunt willen dat je beoordelaar een super expert is, maar die bestaat niet. Dus er zit altijd een stukje kennis die je in de beoordeling niet meeneemt, omdat het ook niet van toepassing is voor je beoordeling.

Sp. 1: Je hebt al uitgebreid toegelicht hoe belangrijk dat ‘willen’ is. Eigenlijk die intrinsieke driver die bij medewerkers moet zitten. Bijvoorbeeld van trainingen; waarom zit je hier? Dat ‘kunnen’, kun je daar iets over zeggen? Als je bijvoorbeeld dit Maturity Model bespreekt in
een team, waar zit dan het ‘kunnen’ voor jou? Kunnen al die engineers in onze organisatie Continuous Delivery…?
Sp. 2: Nee. Dat vind ik ook het moeilijke. We hebben zo’n mooi DNA profiel geprobeerd te maken van een engineer binnen IT. Het is een schaap met vijf poten. Het ‘kunnen’ zit ‘m in het ‘willen’. Het klinkt heel theoretisch en een beetje filosofisch zelfs. Een mooi voorbeeld is X. Ken je hem?
Sp. 1: Nee.
Sp. 2: Die heeft. Die zit op. Hij zit niet in de ‘wil’ mood, hij zit altijd in de weerstand mood. Ik vind het utopisch om te denken dat ik een engineer kan vinden die goed kan developpen, java kan scripten, goed kan testen, het nog leuk vindt om te testen terwijl hij het ook heel leuk vindt om java te scripten. Die bestaat gewoon niet. Kansloos. Wat ik wel belangrijk vind is dat ze willen dat ze een ander willen kunnen helpen. Dus op het moment dat ik een stuk javascript bedacht heb, dat ik het voorkom dat ik een stuk javacode over de bühne flikker en aan een tester geef met de opdracht om maar heen en weer te gaan testen. Nee, dat ik voorkom dat hij fouten tegen komt. Dus ik ga hem helpen door bijvoorbeeld een test zodanig neer te zetten dat ik tachtig procent van de integratie met de rest van de applicatie al opgelost heb. Daar heb ik wat voor nodig, dat klopt. Maar dat niveau is willen. En als je dat wil, dan kan je het ook. Dan help je de tester. Maar die tester moet ook willen helpen, dus het zit ‘m in het willen. Je moet ook kunnen java scripten. Waarom? Omdat als er een foutje tegen gekomen is, kan hij het over de bühne gooien in z’n team. Dan zijn we nog steeds overdracht aan het doen, dan gaan we zelfs de code aanpassen en dan zeggen we: “Kijk eens, ik heb het aangepast en ik kom in dezelfde kit als waar de developers ‘m hebben in gezet”. Maar dat is willen. En als je het wil, kan je het ook.

Sp. 1: Hoe het hele strategische landschap eruit ziet,
Sp. 2: Nee, dus op die manier die brug slaan tussen wat ze kunnen en wat ze willen. Je moet
niet van iedereen hetzelfde verwachten, dat gaat niet lukken. Er zijn hier jongens die kunnen
echt onwijs goed coachen en er zijn jongens die onwijs goed kunnen vertellen wat je moet
doen, dat is een groot verschil. Dat is de kracht van ……, denk ik. Het geldt hetzelfde voor
zo’n model. Je moet niet van iedereen hetzelfde verwachten. Maar van gezegd, van een
scrumcoach, product owner en teammanager mag je verwachten dat ze dit kunnen. En ik
vind dat je van een scrumcoach en teammanager ook moet kunnen verwachten dat ze het
crullen naar engineers toe die hier meer moeite mee hebben, net zoals ik er moeite
mee heb om een code goed te kunnen lezen. Dat vind ik in mijn beleving nog de
ongewaardeerde rol van management en scrumcoaches. Ik vind dat de managers nog
veel te veel op de techniek worden geduwd. En veel te weinig op het coachen. Echt veel te
weinig. Als ik zie wat voor een kloppers hier rondlopen…

Sp. 1: Ik heb nog drie vragen over. Heel algemeen weer over een Maturiteit model als deze.
Als je ‘m aan zou raden, waarom zou je ‘m dan aanraden aan IT-managers in het algemeen?
Dus niet specifiek voor onze organisatie, maar in het algemeen.
Sp. 2: Wat ik zeg, je moet jezelf willen kunnen meten om te kunnen verbeteren. Dus wat voor
Maturity Model je ook gebruikt, meten is weten en weten is verbeteren. Het maakt niet uit
met wat voor model je het doet. Oké, het maakt wel een beetje uit. Het moet natuurlijk
aansluiten op wat je aan het doen bent, maar toch.

Sp. 1: Heb je nog speciale dingen ervaren met dit model, nu je aan de gang bent? Dingen
die je voor dit interview nog niet had bedacht.
Sp. 2: Nou, zo’n interview geeft ook altijd wel weer stof tot nadenken, want je bent aan het
nadenken. Vervolgens ben ik het weer aan het vertalen naar wat ik straks tegen m’n team ga
roepen. Dus dat is altijd wel goed. Is ook weer zo’n voorbeeld; ik ben er mee bezig, leer
ervan en ga het gebruiken.

Sp. 1: Mijn indruk is soms dat er voor jou niet echt nieuwe dingen zijn uitgekomen.
Sp. 2: Nou, nee. Maar goed, dat zou ook heel raar zijn als je al twee en een half jaar in een
DevOps cultuur …. 

Sp. 1: Dat weet ik niet.
Sp. 2: Eureka is één. Laat ik het zo zeggen; ik leer er wel weer van. Zo’n gesprek voeren en
erover nadenken zet je aan het werk. Dus in die zin kun je er altijd wat mee, maar dat stopt
niet. Het stopt nooit. Het verbeteren stopt nooit. Tot gisteren hadden we nog niet bedacht dat
wij als bankbedrijf bijg data gingen gebruiken. En morgen is het al achterhaald. Iedereen zit
nu op Linux en overmorgen denk ze: “waarom doen we dat?”. Het stopt nooit. Als je nu aan
mij vraagt: “is het eureka?” Nee. Dit is dit: “wow!”, maar die begrijpt
er geen zin van. Dus in die zin is de stap tussen een waterval en zo’n model wel heel groot.
Maar als je erover na aan het denken bent, zou het wel eens een eureka kunnen zijn. “Oh,
dat is handig! Daar had ik nog niet over nagedacht. Dat kan ik wel gebruiken als handvat en
als model”. Een Maturity Model is een handvat waarlangs je gaat meten. Het maakt niet uit
hoe het eruit ziet. Je moet het wel een beetje de taal praten van wat je wilt bereiken.

Sp. 1: Heb je nog iets wat verbeterd zou kunnen worden in dit Maturity Model? En dan
bedoel ik het model alsof het zo’n soort plaat is met een uitleg daarbij.
Sp. 2: Ik zou jou willen adviseren. Er zijn meer modellen, meer Maturity Modellen, die niet
alleen maar over Continuous Delivery gaan, maar die wel met softwareontwikkeling te maken hebben. En ik weet, bijvoorbeeld, dat als je een monitor in kijkt, dat het toch voor m’n gevoel relatief veel gericht is op development en minder op operations. In Continuous Delivery vind ik dat wel essentieel.

Sp. 1: Dus je wilt eigenlijk ook meer nadruk op de monitoring capabilities?

Sp. 1: Maar hoe zou je dat zien? Is dat dan maintenance reductie wat je terug wilt zien? Want dat is dan een verbetering die ik je hoor zeggen?
Sp. 2: Nou, niet alleen maintenance reductie, want zo klinkt het alsof het allemaal maar minder moet. Maar, als voorbeeld, over die config managers… Dat soort werk wil je automatiseren om je developments voor kunnen doen.

Sp. 1: Dus het automatiseren van je Ops?
Sp. 2: Ja, daar zit je maturity ook in. Daar moet het ook uit blijken. Op het moment dat ik een applicatie direct kan monitoren, vanaf het moment dat ik ’m deploy en tijdens de development zie dat ik een incident heb. Dat is Continuous Delivery. En op het moment dat ik een incident implementeer. Ik heb een productie issue, ik heb eenchange, een incident dat is geïmplementeerd, ik kan direct terugdraaien en direct zien door m’n monitoren waar de fout zit en ik kan ’m direct oplossen binnen een splitsecond. Dan ben je honderd procent mature en development als het om dat stukje gaat. Natuurlijk gaat het over het hele voortraject, en hoe voorkom je dat? Althans we moeten ten alle tijden proberen om het te voorkomen. Dat is net zo goed maturity en net zo goed een incident wat je implementeert en wat je niet wilt. We doen dit, om kwaliteit te waarborgen. En ik mis een beetje dat laatste stukje in dit soort modellen. Ik weet niet of dat in dit model echt mist, maar als je het benoemt… Mijn punt met dit model is; als je ’m zo gaat implementeren, gaan je engineers er niks van begrijpen. Dat is een beetje de kritiek op die model. Moet je hem daarvoor willen gebruiken? Dat weet ik niet. Maar als je het met je engineers wilt bereiken, moet je een vertaalslag gaan maken.

Sp. 1: Wat in dat model zit meer onder het kopje Information and Reporting, dat is de validatie van je business hypothese, maar dat is precies wat je zegt: vooral development gericht. Dus dit is echt een goede aanvulling waar rekening mee gehouden moet worden. Ook vooral in de organisatie zoals ik ze ken is dat heel belangrijk, denk ik.

Sp. 1: Dat is eigenlijk ook…
Sp. 2: Dat is ook maturity level hoor. Absoluut.

Sp. 1: Dus die zou ik omschrijven als gelijke management rapportage door de hele organisatie heen, automatisch eigenlijk.
Sp. 2: Ja. We gingen minder documenteren, want we gingen DevOps doen. Maar ondertussen zie ik echt zestig pagina’s rapportages elke maand langs vliegen als het over LCM gaat. Terwijl we een tool hebben. Als ik wat wil zien, druk ik op die knop. Dan zeg ik: “wees een man, je bent goed bezig. Je bent echt tachtig procent aan het wegpoetsen geweest”.

Sp. 1: Ja, nou, die moet nu weer aan allerlei powerpoints…
Sp. 2: Die wordt nu vertaald om mensen twee dagen bezig te houden. Dat vind ik zonde. Echt zonde. En ik begrijp waarom hij informatie wil, want wij moeten natuurlijk weer statements gaan geven. Want die worden weer in een riskreport gedaan en hoe minder risk we hebben, hoe minder geld we op de balans hoeven te zetten. En dat geld kunnen we weer gebruiken voor leningen. Dat weet ik allemaal, daar gaat heel veel geld in om. Dus je kan zeggen: “er gaat zoveel geld in om, dat het belangrijk is om te doen”. Dan nog is het een verbetering waarvan ik zeg: “je kan er nog meer geld in om laten gaan. Als je dit automatisert, hoef je het geld niet meer uit te geven om jouw rapportage mooi te maken”. Dat vind ik net zo goed kleine beetjes, maar dat is ook agile. Honderd kleine beetjes is heel veel.

Sp. 1: Oké. Ik heb de antwoorden op al mijn vragen.
Sp. 2: Ik vind het leuk om te doen.
**Interview 2: CD Expert & Coach**

<table>
<thead>
<tr>
<th>Interview 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of minutes:</td>
</tr>
<tr>
<td>Number of speakers:</td>
</tr>
<tr>
<td>Language:</td>
</tr>
</tbody>
</table>

Sp. 1: Thank you for participating on the interview. My first question would be: how important is Continuous Delivery for your organization and why?

Sp. 2: For me Continuous Delivery is very important. The main reason being we are moving into a much more digital and apps enabled age where everybody expects to be able to go online and find the answers they want or use their mobile device to find the answers they want. Being able to respond quickly to that demand is something that requires much more reactive, instant delivery capabilities. So understanding what your users want and providing that. It will have to be done in much shorter time frames. Continuous Delivery for me is the key in of that capability. Continuous by definition means ‘all the time’, frequently, based on what your users are demanding. It will be super important and those organizations which provide it, will survive. Will be in the front of mind in the marketplace, people will use them. Others, who are slower to react will effectively have issues and potentially die.

Sp. 1: Thank you. That was very clear. The second question is: what is your experience in the usage of Continuous Delivery Maturity Models?

Sp. 2: This is why I feel like I have a confession to make. I have worked… In fact, I’m a participant of an organization who runs exercises with clients for doing Continuous Delivery maturity assessments. So I’ve done a few of those for that organization. So I have experience of a particular model of taking that out to clients, organizations and trying to apply it. So doing the assessment work benchmarking where they are on those dimensions with those skills and provide in reports on how they potentially could improve. So I do have a bit of an inside. Not of this particular model, but you’ve presented with me. But with variant of a similar theme.

Sp. 1: Ok, thank you very much. What should be the main added value of using a Continuous Delivery Maturity Model?

Sp. 2: The main value is being able to understand, first of all, where you are on that scale. Which can be important to understand how you compare potentially with peers in your sector or in your marketplace. But also to map out that route towards better maturity. I don’t know if that’s the term you use, but becoming more mature over time. So having that road map, that planned action which says: here’s where I am now, here’s where I need to be and here are the steps I’m going to take to get there. That’s the key value for me.

Sp. 1: Thank you. What are the Continuous Delivery focus areas most relevant for organizations by working Agile/SCRUM in DevOps teams and why would you think those focus areas are the most important?

Sp. 2: Ok. As I said, I’ve worked with many organizations in this space. It does really differ depending on the organization you are working with. Organizations focus on different areas, some have an organizational discrepancy where they are and have no communications between teams. There the focus area is on team composition reorganization by getting the right people together. And some have tooling deficiencies. So it really does depend on the focus area of that particular organization. The question I was going to have back was… You have asked about agile and scrum. And agile and scrum is like a method of delivery of
Continuous Delivery and it doesn’t need to just deliver Continuous Delivery. It could deliver several services or anything really. And that has an objective and something that has to be delivered. So what I am seeing more and more is businesses using a scrum type agile framework to improve a particular focus area. For example, on the organization aspect of having cross functional scrum teams. Or if they are looking at tooling and they want pilots and doing a proof of concept for a tool, having a scrum team with a fixed backlog, time box iterations prove that to. So you’re getting regular feedback on how things are working. For me agile and scrum is totally complementary and probably the best way of delivering Continuous Delivery change. Again the change is depending on the focus area. Does that answer your question?

Sp. 1: Yes. Question five: what critics do you have on CD Maturity Models and why? If you have any.
Sp. 2: There are two main criticisms. Probably well known, but I’ll reiterate here then. The Maturity Model is always somebody’s perception out a particular point in time. It’s all based on where we are today. So your definition on what may be good, better, excellent or whatever the tiers of an experience are. They are based on somebody’s subjective opinion. So depending on who develops the Maturity Model, and can be developed by many attributors. But they are based on a point in time, which obviously then changes over time. So as the industry gets more mature in Continuous Delivery, for example, you may be an advanced level today, but in a year or two that might actually change to an intermediate level or you’ve made an intermediate to a beginner level. Because of the advances made by the industry, peer group, etc. So they do need to be revisited in terms of ensuring that the dimensions are correct. And also you then need to continue to find the model and find what the new expert level is. Because as the expert level becomes the norm, which is again what happens, the contributors to the model need to find out what the next higher level is. The main criticism is that they are a snapshot in time and it is difficult to continually find out the “what is pushing the boundaries”. What is the leading edge capability?

Sp. 1: Ok. About the subjective… you’ve worked with Maturity Models, can you say something about how that will work in practice? For example, the users of the Maturity Model, are they really overestimating themselves? Or underestimating? Or is it clear?
Sp. 2: Well, that’s a thing, because you are trying to assess both individuals, I suppose. Because you have conversations with individuals in the organization. But in the organization is a hole, you do get mixed answers. The people talk about the way they like to be, rather than the way they are. So they paint a more rosy picture. Trying to get a higher level. Because usually what happens is this kind of assessment, maybe used at a management level. And people are sometimes a bit nervous to paint the picture of themselves as being the lower levels, in case there are repercussions. The way you manage that is you just explain that the business as a whole wants to improve. Let’s find out where you are and have a practical program for approving. But it’s human nature to try to beef yourself up a bit, trying to increase your capabilities. So be very aware of that happening.

Sp. 1: That’s clear, thank you a lot. These were more my genetic questions. Now I would like to focus on a specific Maturity Model that we can look at the same time. So how could this Maturity Model be helpful to determine the status of maturity in an organization?
Sp. 2: I’ve worked with a couple of versions of this kind of model, so this is a new one to me. But the key are really are… When you’re looking at an organization, it’s usually people, process and tools. There are three key areas of focus. And I think the model you provided here let you evaluate the people dimensions through things like culture and organization. So there is that capability. The process side of things is covering in terms of a number of the other tranches of the model. I think that you have the chance of fully evaluate all of the processes. The tooling side of things, I think, is embedded as well within a lot of these dimensions around design, architecture, build and deploy. It does describing having tooling in place to do that. At the requisite dimension. In terms of applying; yes, this would be very
helpful. And the information of reporting one in particular is quite a new one, because a lot of views on Continuous Delivery is that it stops at delivery. Where actually it doesn’t, that’s a bit of a myth. It should actually go beyond delivery into proactive monitoring, self-fixing/ self-healing situations, because as you automate everything and you have an enough fully automated deploy model, you should also have a fully automated rollback model, which is based on things like information and reporting. So like checking the health of your environment. When you introduce a new element through Continuous Delivery, like a new component or a change or a fix. If that has a knock-on effect anywhere else in the whole eco system, you should be able to roll a lot back. And you can only see that by having adequate reporting information. It’s a bit like the human body; you do some on one part and assume there is no impact on the other part and you don’t test that, you only test a couple components that are close together. And by having more effective diagnoses and monitoring tools in place, you see the wide repercussions of making changes. And you need them to heal those.

Sp. 1: So in the interlink between those people, process, technology kind of topics?
Sp. 2: Exactly.

Sp. 1: Ok. Clear. Question two is: how could this Maturity Model be helpful to determine the next steps of implementing improvements in your organization?
Sp. 2: It does give an indication as in what the next steps are by… If you established where you are in this particular dimension, you can see some gates to become. To move from intermediate to advanced, you have to focus on the gates on the advanced level. So that would give you a specific set of steps you could follow. An example, being in build and deploy you may be intermediate doing automatic builds and deploys, but you may want to move to an advanced model of doing zero downtime deploys. Therefore, what are the steps in your organization that need to take place for that to happen? But again, there is a repercussion the whole model, because there may be things in the build and deploy strand which require support from other areas. So it’s important that the model shows the interdependency between the areas. So doing zero time deployment requires, for example, culture and organizations changes. So it’s important that you consider that dimension as well. You can’t consider the dimensions in isolation.

Sp. 1: So you’re also saying that it would be strange if you have a level one on one area and a level five on the other area? Meaning one really base and the other one really expert.
Sp. 2: Exactly. You attend to see some cohesion between the two. But again the point there is that if you want to move up in one dimension, you need to take the other dimensions with you as well. You can’t just focus on one in isolation.

Sp. 1: Thank you very much. Question three: what should be extended in this particularly Maturity Model and why?
Sp. 2: Extended as in ‘how would this be extended’?

Sp. 1: Yes, do you miss something in this model?
Sp. 2: It’s sort of related to an earlier point I was making. This again focuses on build and deploy. And it does mention zero touch continuous deployments, the expert level, which is fine in that dimension. But coming back to the information and reporting side of things and being enable to monitor and self-diagnose where deployments break things. I would maybe expect to have a strand on…

Sp. 1: Category.
Sp. 2: Yes. Environment status health check or environment reliability. So how reliable is the environment. Because in a lot of organizations the environments are manually managed, paged, alerted, things are happening, pulling their hair out. So based level where the expert level may be full self-healing environment, monitor check, and roll back deployment changes.
Maybe something around the environment status would be a useful dimension. That was the only one I could think of when I was reading through this one. But again also coming back to my earlier point, what is beyond expert and how to maintain that continuous new view of…

As an expert becomes advanced for most organizations, what is the next level of experts? And having somebody looking at that.

Sp. 1: Ok. Question 4. What recommendations can be given next to the Maturity Model to implement Continuous Delivery efficiently and effectively in your organization and why? I know it’s a broad question, but what comes up? From a supporting the Maturity Model or supporting Continuous Delivery perspective.

Sp. 2: When all of this supports the Continuous Delivery agenda, but again it does depend on the organization. Because some organizations can be coached as in; saying due to DevOps and having DevOps teams, they may not have some of the other dimensions in place, which is doing the test and verification ability. They may be doing very good in terms of a new team dynamic, but if they are implementing for automated test and verification, they are never going to get Continuous Delivery/Continuous Deployment. So it’s about making sure that you have all of the dimensions relevant within your business. You can’t really choose one in isolation. And for an organization you have to take the full model, it’s a bit like doing scrum. For example, people say by doing a stand-up, they are following the scrum model. That’s just not the case. A model works when it’s applied in it’s whole. When you only apply a piece meal, then you only get part of the benefit. You never get the full benefit. So for an organization you need to apply this comprehensibly. And again it can be a lot of work for organizations, so you do need their management buy-in to support. That side of things.

Sp. 1: Thank you. I think the summary is that the recommendation is to really use the full model and not making the mistake to only focus on one part, because that will probably not help. And the management buy-in is also really important. And I would also say: make it really fit for your organization, because it is a generic model.

Sp. 2: That’s the thing with a lot of these models. You have to translate them to an organizations own language sometimes, because every organization has that. The people might not call things ‘deployment’, they might call them ‘installations’. You have to tailor the model as it’s being applied to that particular organization. You wouldn’t need some information on the organization, there will be some pre-requisite research required. So you understand that the organization a bit better. So don’t know if there is a question on this, but the assumption that you can make is: apply the model on day one might be wrong. You better spend a little time with the organization first; understand that the dynamic, understanding the language, the people involved before you could apply the model. So just be aware of saying…

Sp. 1: Send out a mail: This is what we do, as of tomorrow.
Sp. 2: Where are you on this model, yes. That kind of thing.

Sp. 1: Good advice. Thank you a lot. The next question is: how will you use this Maturity Model for self-assessment in the future and why or why not?
Sp. 2: It would be used. I do continue to use the models that I’ve worked with in the past and I try to keep them visible physically in term of print outs or at least visit them regular on my laptop. Just so that you’re aware of where you’ve come from and where you’re going to. Because when you come to an organization, for example, you might sit there and say: “oh, you have a doing automatic integration test”. So there is an assumption there that there are also automated unit tests. So it’s useful to just double check that kind of thing. But it’s also important to know what is next. So doing the automated, but isolated component test. And doing automated acceptance tests. By having that front of mind, for your own self-assessment is useful to know where you’ve come from and where you’re going to. And the kind of things to keep an eye out for. And the kind of recommendations again you would
make to your organization or your peers in terms of how they would become better. I assume a self-assessment is, as an individual, how to make you better than where you come from.

Sp. 1: Ok. Why would you recommend this Maturity Model to other colleges, IT-managers who are working inside or outside your organization? Or why not?
Sp. 2: All models have a place to play and this particularly one has a good broad mix of both dimensions in terms of what you’re focusing on and the levels of maturity. So I think some models are a bit less granular than this. I’ve seen a model that just has three columns with four levels maturity. This one seems a bit more comprehensive. So in terms of recommending it lies a bit more flexibility for organizations to mop themselves onto this. Also models may be a bit constrained, this allows a bit more flexibility. The only concern I have again is around the language of the maturity levels. To use things like beginner, intermediate, advanced, expert. You’re making some subjective. You’re giving some descriptions there that some people may not be comfortable with.

Sp. 1: What would you propose then? One, two, three, four, five?
Sp. 2: Yes, or A, B, C, D, E. People always say what’s the highest…

Sp. 1: You could change that also?
Sp. 2: For me numbers could work. But again the issue there is: what do you do when you get to five? An organization might be doing all of this very effectively. They could think: “where is six?”. Where do you stop? I suppose you have to make some judgement call somewhere on what you call the levels. That’s perhaps the only thing that may make it difficult to recommend this particular model. But again if a model had numbers; what’s the five? What’s the maturity level? People still weren’t comfortable saying they’re a one on dimensions, because it makes them look low. So I don’t think there is a right answer to that.

Sp. 1: One is better than a zero.
Sp. 2: Yes. I suppose you could start with doing ten, twenty, thirty, forty, fifty, so you’ve got flexibility.

Sp. 1: Ok. Thank you. Is there anything in particular that you’ve learned from practicing or reading about this particular Continuous Delivery Maturity Model?
Sp. 2: Yes, it’s useful for this particular model compared and contrasted to both my own understanding of what is beginner, intermediate, advanced and expert. But also to see what expert really is, because again that’s the ultimate end goal for a very mature Continuous Delivery organization. So you should make sure that is much as both what my own understanding and how it compares with other models. Things like “infrastructure as code”, I think is considered as an expert level on this. So my opinion on that is that it may not be an expert level, because that’s something that’s being mainstream for quite some time now. So that would be at best advanced. Again, that’s the thing with all of these models. Somebody somewhere has made a subjective guess of what they think and it’s useful to use this to compare with others. Ultimately to form your own understanding of what the maturity levels are and what is expert.

Sp. 1: Ok. My last question is: what could be improved in the Maturity Model? I think you already gave quite some improvements.
Sp. 2: Yes, add environment health, but also how you extend to be expert would be my main improvements.

Sp. 1: Are there other improvements?
Sp. 2: The one question I had around the model is: in your application of this, would you give an organization as a whole a maturity level, as in based on beginner, intermediate, advanced, expert or would it be based on individual dimensions? Because I’ve worked with models where there is one end score. You are 3.5. And that 3.5 is based on things like
weighted scoring on these things like culture and organization, because we know in terms of DevOps and Continuous Delivery, that’s one of the hardest challenges, but also one of the biggest enablers of change. So the scores in those levels get rated higher than some other more routine tests like writing a unit test. So the model may require… It didn’t come out in the documentation, but whether if you were scoring somebody on this whether there will be weighting add it for particularly areas, which will automatically give you a single score.

Sp. 1: I don’t really know. I think to believe that the aim of the authors was, to also be able to use the model for an enterprise. And I fully agree with you that and extensions could be granularity in as improvement. And I think, what was also your feedback, that, for example, when you take these kind of Maturity Models and you discuss it with certain people. Like managers on different levels in the organization, how to map. Do you map where you see the lowest Maturity as the Maturity or do you map in a scored/or weighted band.

Sp. 2: You would apply this separately within an organization, because every organization has his own internal differences. And there will be some teams who are really proactive in this sort of space, who might have an advanced level. And there will be others who you have to drag with you, they might be based as beginner level. So it’s difficult to come up with a single score for the whole organization. It may be that team in the organization have different scores. That can be a good thing, because it could generate healthy competition for most of the teams. Of course it has his improvement points, you can say: “you’re a beginner team, spend some time with the expert team and try internally. Spread the knowledge”. So that might be…

Sp. 1: Yes, that’s a good other usage of the Maturity Model.
Sp. 2: Yes, you can have that lead table of who is the best. But that might work both ways and people are against that. I suppose I can describe it like that. But I think I can spot that certain part of the organization that can help others!, that can only be a good thing.

Sp. 1: Ok. Thank you very much. Do you have any other comments or remarks to make?
Sp. 2: No, I think in terms of Maturity Models… I have worked with them and applied, so I do see the value. But they need to be and that maybe feedback for the authors. They do need to be continually refreshed to reestablish what are the new levels of maturity. Because what is expert? The expert of today won’t be the expert of tomorrow. That all needs to be refined. But it was an interesting discussion. There are a few models out there, so it’s interesting to see another variant of that. They’re all broadly the same in terms of content, but sometimes the levels are different. But again… Is this a Dutch model or is this a UK model or is it from the US?

Sp. 1: Three experts made it and one of them is from Sweden. And it’s published the internet. So it can be accessed easily. It was published in February 2013 and according my information, it has not been updated since. This with regards to your comments on the most advanced levels. Ok, thank you a lot.
Sp. 2: Yes, this was useful.
9.3 Interview 3: CD Expert & Coach

<table>
<thead>
<tr>
<th>Interview 3</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of minutes:</td>
</tr>
<tr>
<td>Number of speakers:</td>
</tr>
<tr>
<td>Language:</td>
</tr>
</tbody>
</table>

Sp. 1: My first question is: How important is Continuous Delivery (CD) for your organization and why?
Sp. 2: I think CD brings much needed change in the engineering capabilities of the bank. Because you can really improve the quality of the code. We can automate a lot of manual processes which are used for deployment or quality checking and so on, and thereby you can deliver software better and faster. So, the CD holds that promise. And that’s what we are trying to do here.

Sp. 1: Ok, thanks a lot. Question 2 is: What is your experience in the usage of Continuous Delivery Maturity Models?
Sp. 2: Yes, so, when we started last year, with CD implementation, than we started using this CD MM and we tried to map different teams to where they are at this moment. And, what we found, the things would not in kind of orderly manner, when we do this, then we do this and then we do this, an all that. Like, for some practices there are based for some practices they were already intermediate and all that. So, for each team, we create a kind of a picture, where you are. Like something like this.

Sp. 1: Yes, ok.
Sp. 2: And then, from there we thought, well that our ambition is that by the end of June, which will be one year, so desired condition. Everybody is at intermediate level.

Sp. 1: Ok.
Sp. 2: That’s what we said. It is possible that you are at advanced level, in some aspects he?

Sp. 1: Yes.
Sp. 2: But, at least all of us have to be in this level.

Sp. 1: Ok. My third question is: What should be the main added value of using a CD Maturity Model?
Sp. 2: I think the real added value of this model is when you are working with a large number of teams, because a lot of number of teams, their using multiple technologies, they have the multiple kinds of solutions, for example: one team is building cold from a scratch he? While other team say they create of say software from a builder and then they customized it and so on. It’s a lot of like these are different or these are working, different team, different qualities. This model we create a kind of one language to talk about the team immaturity.

Sp. 1: Ok.
Sp. 2: And that’s what this model, that’s how this model really helps us.

Sp. 1: Ok.
Sp. 2: So otherwise it will be impossible to know in all those 45 teams start operating in our organization were they are on their and let’s say road to continuous delivery.
Sp. 1: Yes, ok, very clear. My fourth question: What are the CD focus area that is most relevant for your organizational scope by working Agile/SCRUM in DevOps teams?
Sp. 2: So I think, if you look at CD we are picking this up as a whole so, we don’t say that this is less important or that is more important, of course in the roadmap of the team they have an option to choose one thing first and then the second thing. But, if you look at where we put the most emphasis at this moment, is first of all, having a uniform definition of done. In making sure that people understand what definition of done is. And then after, we slowly try to automate so the definition of done can be enforced more and more using automatic tools like iValidate.

Sp. 1: Ok.
Sp. 2: So, if you look at what is particular focus? I wouldn’t say there is one focus, it’s like all the areas are important, but I know there are some teams are putting a lot of effort in automating the functional tests. They are investing a lot of time and effort in that. There are other teams they are spending more time in reducing their technical debt and improve quality and so on. While some other side of the teams, they are quite busy, in automating the deployments. So, if I say I am supporting a vendor product, and I am not doing coding here. So, for me it is important that I can deploy this product very quickly on D, T A and P. I guess that there eventually everybody has to come to the same level and this is our ambition and this is intermediate level.

Sp. 1: Ok, really clear. My fifth question: What kind of critics do you have, could you have on CD Maturity Models and why?
Sp. 2: I don’t have any particular criticism. But what you see here in this, this is not a criticism, but more like a remark, if you look at this CD MM, it contains technical and not technical things he?
Sp. 1: Yes.
Sp. 2: So, for example, the, here, the things here that "prioritize work"? The way we have designed this program in our domain, that we have separated CD from the Agile Scrum. So “don't prioritize work” is important, but how teams are going to prioritize to work he? That we don’t enforce on the CD side, but our Agile coaches assist the teams and they decide that. But, if you look at this model, this is kind of little bit....

Sp. 1: Mixed?
Sp. 2: technology and other things mixed in it he? Especially if you look at the culture and organization part, of course it is essential for CD, but, from the CD perspective, it don’t influence it technically.

Sp. 1: Ok, clear.
Sp. 2: And then there are a couple of things in this model that basically do not suit let’s say our, do not let’s say fulfill our vision. For example like: a dedicated tools team he? Now, do we have a dedicated tools team and I do not know how to really map this to our situation. So, from time to time you do some things that you really adapt, but otherwise I think it is a fine model and we are being using it properly.

Sp. 1: Ok. Some questions that I have on the model is: How, and there is some overlap in the questions, so sorry for that: How is this MM helpful to determine the status of CD maturity in your organization?
Sp. 2: How did it help?

Sp. 1: Yes, how do you determine the status of CD?
Sp. 2: I think, per team you can really clearly measure where they are. The better they are base or intermediate.
Sp. 1: And let you do the team do it, or is it somebody else that comes to the team to do it, or...?
Sp. 2: Yes, we do this together with the team, the coach sometimes, we sit together, and then they do this measurement.

Sp. 1: Okay. Thanks. The second question is: How is this MM helpful to determine the next steps of implementing CD in your organization?
Sp. 2: We are not using this model to measure the next steps, we know as well what we have to accomplish with Continuous Delivery, Continuous integration, automatic deployments, and automated monitoring. And automated environment provisioning? Now, mostly the automated environment provisioning lays outside the teams, so, the others on ING. So teams are having to work on CI and CD and automating monitoring, these three areas. Other than that, they have the freedom to decide what they want to do first, as long as they reached the same goal. In what order they achieve it, that is not really important.

Sp. 1: Okay, thank you. Looking at this model: What should or could be extended in this MM?
Sp. 2: Now, I don't look at it from that point of view. About the completeness.

Sp. 1: What comes to mind? I would say, especially the categories would you say like he, I miss something here?
Sp. 2: No, I am not sure. Maybe I'll have to take a fresh look at it, but I, so far I have not found that if anything is missing.

Sp. 2: Yes. Maybe they can look into the level of abstraction in it, because there are a few things in this model which are really let's say: small things? And then there are a very view things that like very large, so for example pulling builts one action will come and then a couple of things like cross functional teams could be something very big.

Sp. 1: Yes, thanks a lot. Fourth question is: What kind of recommendations next or outside the MM could you give for organizations for implement CD efficiently and effectively.
Sp. 2: Yes, so, I think that....

Sp. 1: Some golden tips I would say.
Sp. 2: Yes, I think that the standardization is very important to standardize your tools. And then making sure that people understand that tool. And they start acting upon let's say the matrix that they get from those tools. Because, when I found that a fast setup of tools, and then really use tools, but sometimes people don't act upon it. And then it's waste. But I would say once IT has the right tools and that everybody has that same type of tools, and then make sure that people start using them. And my experiences, there to making people leverage those tools if they don't have CD background, that's very hard. We keep pushing them and pushing them, than something has happen.

Sp. 1: Okay, thank you. How will you use this maturity model for self-assessment in the future and or why not? Think you already said you use a similar....
Sp. 2: We are using it and I think we also combined it with another one which is the CD Maturity checklist, from years on. We are a little bit over there. They filled in the numbers and aggregate number comes out.
Sp. 1: Ok, sixth question is: Why would you recommend this MM to other IT Managers working inside or outside your organization? We are really focusing on IT Managers. Would you….
Sp. 2: Why I wanted this model is because I think this is a quite comprehensive weight, I think it covers much more than CD and it condense everything, so you can tailor it to your needs and all there. So, I you search for these kind of models, then there are a very view models of this level and this is a good one so everybody can understand and realize so that is good.

Sp. 1: Ok. Thanks. And my seventh question is more also from an IT management overall perspective: have you learned something from practicing with the CD Maturity Model?
Sp. 2: Not as such. Not because we used it as a reference, we did not use it as a learning tool, a lot of learning help us, so on the work floor where the real experience are the people.
And not from more or less.

Sp. 1: Ok. And my last question is: What could be improved in the MM and why to a bit similar as the extension?
Sp. 2: As I said direct to the, it is better tool, the abstraction level is good. I think maybe it will really help if they can make these, make it easier to measure these things he? Because what I’ve found is that the, when you interview two different teams, the, about, let’s say one practice let’s say configuration management he? There could have been a lot of distinction of maturity between these practices or the two teams. They might both think that they well are big enough because I am doing it he? And then how do you really determine that who is at that level and who is on that level he?

Sp. 1: Yes.
Sp. 2: Yes. I think if more help about interpretation of these things, that would be really helpful. Also for example, this question he? This one is the one-liner API management he? Now, what do you mean by API management? The, and how do you really conflict real situations, all those things that can do probably they can explain much better, I think that would be very useful. Like version control, database changes he? This…. in our situation we found that this probably for us, will happen here. Not in beginner stage.

Sp. 1: Yes.
Sp. 2: But, only if it were very difficult, to version control db. changes, and so all these kind of things.

Sp. 1: Ok really clear.
Sp. 1: Ok, so my first question is: How important is Continuous Delivery (CD) for your organization and why?
Sp. 2: Well, I think Continuous Delivery is very important for our organization and also for our area Financial Markets and even more specific for our, let's say L3 area. And it is important in order to improve the time to market improve the quality, and make also work more fun for people to move away from a lot of manual work and the boring work. Work that is not fit for high performance IT workforce. So yes, very important.

Sp. 1: Ok, thank you. What is your experience in the usage of Continuous Delivery Maturity Models?
Sp. 2: Yes, well I seeing some models here and there, but the be honest really using them, that's not something at least I did, consciously.

Sp. 1: Ok.
Sp. 2: Thought about it in steps, in different areas, take for example the testing areas, the deployment area, the and even the culture area, but never the really structure I say it model, there were some more dashboards at one point I think I never use those.
Sp. 1: What is in your view, should be the main added value of using CD Maturity Model?
Sp. 2: What should be the main added value? Well, I think the main added value would be to provide them you know, clear direction and clear steps. Towards continuous delivery consistent across the and implementation of CD. And also aligning I think would really help in aligning different teams, in different parts of the organization, to have something that is just, it is whole organization.

Sp. 1: Ok, thank you. And what are the CD focus areas that are most relevant for your organizational scope so by working Agile/SCRUM in DevOps teams?
Sp. 2: CD focus areas. Well, I would mention the deployment side, so the automatic deployment part. I would say the, does it has to be according to the model?
Sp. 1: No, no, can be.
Sp. 2: Yes ok, but then I will say, I will say really the deployment part, the testing part of the automatic testing. Then I say automatic testing, I am talking about regression testing, so let say really from more, even more business point of view. But, also the performance testing which is also very important, what else? CD focus areas I would say also critical, really critically is the culture, and how people are working with each other, so, that is also important.
Sp. 1: Is that more important as the, do you believe as the deploy- and test category?
Sp. 2: Yes, they kind of go hand in hand it’s very difficult, but I would say in the end yes but, how more important? Anyway, I would say yes, because in the end they are the people that are using the tools, so the moment you have the right kind of minds, and the right kind of approach, the right kind of interaction, than you know, that will also trigger for sure later on, the development of testing and tools and some, but in the end it is about the people using the tools. And that’s, I would say, from our organizations scope I would say that the mindset the shift that is needed is the most important one.

Sp. 1: Ok. What kind of critics do you have on CD Maturity Models and why?
Sp. 2: As I said, I didn’t use any, but I did read about some. I think somehow there are, sometimes they are just complex and they seem complex, they seem overwhelming, and in a sense of be so much to do.

Sp. 1: Yes.
Sp. 2: Not in difficult to understand, but it is... I don’t know, it is just seemed just a little bit complex. But, and also I would say they don’t always seem to fit the organization, of course elements are fit to the organization, so, but then it’s is not applicable to us, this doesn’t work, it is not relevant, so than you are a kind of losing confidence in the model even though the model could be more defied adapted for your own area. I think that’s the first, every time I read something I was like: wow, that’s another big theoretical, how do you say it, exercise. And now I do: what’s next? What do you do? So, it’s the kind of applicability of it to the specific situation, to this specific moment, you are in. So I think that will be the big one, but, still, not working with one on day to day basis and not one I said bash to read the MM is too much.

Sp. 1: Ok, thanks. Than I have some specific questions on the CD model of Rehn. First question is: How is this MM helpful to determine the status of CD maturity in your organization and why?
Sp. 2: Well, I look through it and to be honest, if we define CD as it is defined here, with this elements, which seems to be correct, looking at the literature and the experience, I would say it’s just, maybe I don’t understand the question correctly, but if you go through the, if you really can go through the point, see where you are, you reflect a bit, put things in prospective, and then say ok, well I am at this level, and I understand where I am in my growth towards expert CD organization, so I would say that it’s seems very helpful, and it seems quite easy to use, at least at first, and yes, that’s what I would say.

Sp. 1: Ok.
Sp. 2: Is it a good answer?
Sp. 1: Yes, it’s a perfect answer. How is this MM helpful to determine the next steps of implementing CD in your organization and why?
Sp. 2: Yes, I would say that the moment this is, a model like this is really understood in the organization, also senior management level, but also at business level, and you can have I think a much more even, yes, I would say almost fact based discussion on where you are on this moment, and saying ok, this is what we intend to do next. It is according to a model, there are many models there, but this is you know, used in the industry. So, then you can easily see you know, what will be the next step, and not jump over steps. That sometimes happens. They seem to be quite incremental steps perhaps, it’s also some you can explain to the businesses and also get you some maybe I think a bit of credibility. This is you know, we are really on this road together, we are getting there and this is where we are, and this is where we need to get and this is the next step. I think that is actually my help with conversations here and there at least.

Sp. 1: So, the structure of it and also the effect you can explain in incremental the steps in different Maturity Levels?
Sp. 2: Exactly. Because now it seems reflecting on my experience is that, I don’t know, it’s we need to get it there and this is far, you know, far far away and how do you get there, what is the next step? And that makes it sometimes difficult also as discussing with the business. So, I think this gives, puts things more in context makes the conversation easier. That’s why I think I never try this. So.
Sp. 1: Yes, thank you. Question 3 is what should be extended in the MM and why?
Sp. 2: Yes, it is a model, and that might be the issue, but I think to make it… yes, it is a model, but I it should have, it should come together with some case studies, I know that it is still… it usually is not a part of the model, but, so case studies to start this some clear examples you know like reference stories where you can understand a little bit better what they need, I think they explain it in the……
Sp. 1: Yes.
Sp. 2: but still, you know, those kind of examples, and give a feel that is difficult, but give a feel of the case duration or how difficult things are, that the main pitfalls, bottlenecks and so on, that’s the practical part, on the real, let say model, so, what’s here on paper, I would say to be honest, it’s more than enough, I mean there’s quiet enough explanation and so on, I would need to think a bit more what logical blogs are missing from here, so, yes I would say from a model point of view I don’t know what exactly to help.

Sp. 1: Thanks for your advices. What recommendations can be given next to the MM to implement CD efficiently and effectively in your organization and why?
Sp. 2: Can I give a recommendation next to MM?
Sp. 1: Just what comes to mind? I know it is really an open question, let’s rule it open.
Sp. 2: Yes, I would say I mean again what I mentioned earlier, you know, the examples in best case practices and also I think, I think it’s really about expert knowledge and we also tried it here to bring in continuous delivery experts and so on, but I think, real, even if that exists, people that had really experience of continuous delivery, also on the tooling side, also on the organizational side, also on the cultural side, I think there is stuff to buy on the market, that is my impression, and to make it more efficiently and effectively, I think looking at the journey until now, it could have helped a little bit more to have people with experience here and we had a few but I think even more could have been efficient and also a bit more common sense making where we agree on what we do, and once we agree, the more has to do to make a decision and move on and then after I would say, I don’t know, two sprints or 4 sprints and come back for a second with: this is what we did and we move again and we kind of comfortably quicker to an, how do you say it? To a way of working.

Sp. 2: And we are a bit too loose so if I, I would say here I would say expert knowledge, from outside, than some structure it is agreed up front?

Sp. 2: So even some steps? Even these steps?

Sp. 1: Yes.
Sp. 2: Planned somehow or at least, some timelines. Let’s put it like that, and maybe not a dashboard in itself but still something that you can discuss about and kind of….

Sp. 1: Something like that? Implementation plan or….?
Sp. 2: Yes, something like that.

Sp. 1: Ok.
Sp. 2: Something that you, at one point, you agree and try to make it, because otherwise you don’t….. you never it seems that we don’t know exactly where we are. We are somewhere for sure, we progressed a lot for sure, but where are we and what do we want to? So, that’s what it is with the model, but I think the model is a good, let’s say way to start.

Sp. 1: Yes, ok. Question 5: How will you use this maturity model for self-assessment in the future and why or why not?
Sp. 2: I was really thinking when I reading it, because now I see some areas that passed it’s is a little bit more technical, and it’s quiet interesting to have a good discussion with some of the experts in the team for example and say: what do you think about it for example? Hey, you see? People are talking, people are saying something, some people are saying, this is the direction, and what do you think about it? Can we do something? So…
Sp. 1: So as an input also to get, to stimulate to talk about it with an knowledge expert and have a discussion?
Sp. 2: Exactly. Exactly. Because what I feel as I said is that people are also really discovering things while there going and this at least can provide us more direction or certain directions sorry. So I, for me, I read it once, I read it I think a while ago, and I saw a couple of them, but I would just you know, just try to understand a bit where we are at my own department and see for myself, at least at first, ok, is there anything that we are scoring really low on and we need to take... make it a bit higher and then on specific topics like building deployment and test and verification, have the discussion with the experts. Which maybe inspired also and saying that it seems that other people who went through this, so that's why we chase.

Sp. 1: Ok, thank you. Question 6: Why would you recommend this MM to other IT Managers working inside or outside your organization?
Sp. 2: Yes, well, I looked at it, and it's quiet easy to understand. If we are talking about this one, the levels are also quite clear discussed, and described and then, I think it's a quiet a clear model......

Sp. 1: Ok.
Sp. 2: and it's has quiet practical steps if you look at the test verification at self-explanatory or even with this piece, it can always go to the other one. But yes, I would, maybe for the next question but.....
Sp. 1: Next question. What have you learned from practicing with the CD Maturity model and why?
Sp. 2: That’s a good question because I didn’t really practice he? With the CD Maturity model.
Sp. 1: That’s fine, just something that comes into mind. Better than nothing.
Sp. 2: No, it’s not.. what I also wanted to say for question 6, for recommending them the model would also say you know, what you learn by practicing, ok, it is a model, but it is also, you need, I think it’s important to understand you know, the underlying logic. And it’s difficult to understand, because it’s a lot to mention as I told earlier. So, I would say you know, just you need to make sense, I would need to make sense of the model, this is the same thing that I would recommend for another manager from somewhere else, and we have some discussions about it with some experts, and then just go and do something about it. Maybe it’s too simple because ok, these are some guidelines, and then you need to execute it, you need to work on it. So, I would say, you know, good to have it as an anchor, and then just start executing and learn about it. So there’s my, that’s my only comment here.

Sp. 1: Ok.
Sp. 2: And it’s the same recommendation I would do for an IT-manager.
Sp. 1: Ok. Any.... The last question is question 8: What could be improved in the MM and why? You already gave some, I would say recommendations for example to also add use cases. Or do you have maybe other improvements that you would like to see?

Sp. 2: I know, I have some online trainings now. What can be improved? I would say in the end, you know, the Maturity Model, it’s as strong as the, how do you say, the implementation plan the implementation approach you have behind it. So, it should be more, maybe it’s more, I don’t know, that somehow more detailed on like cheat cards you know, like the big book, of CD at that level you can make it more practical and you can have the right discussions for people. The don’t need to know exactly what’s there. But each square in the model should then have some as I said a case study. An explanation, and so on and then it’s quite a lot, but then of course you also takes a part for a year, of a year and a half, maybe two, to how is to get there. And you should take it square by square it should already give you enough information that you can choose and pick and say: ok, this is a kind of something that, to be for a specialist to discuss, this is something to take in to discuss with the business, this is something that I can bring in as information these are two, three case studies that I can use in my arguments towards the special reluctant specialist or a reluctant business partner, but then it’s quiet a big pack but I think that kind of pack would be quiet useful especially for big organizations to get stuff done. So, that’s what I would say, but as I really talk about just modelling itself, it’s good enough.

Sp. 1: Yes, but you say the model I would say if I would rephrase it, the model alone is not the Holy Grail, you need, let’s call it: implementation plan, let’s call it use cases, let’s call it certain other structures to really make the journey.

Sp. 2: Yes, I would say so. I think, if I think about it, this looks like something that would give let’s say, you have 15 to 20 consultants, really something to do, and really work hard on it, because it’s quiet difficult, other than that I think we do a lot of re-inventing the wheel. And I think some of it is good, but some of it is not needed. So, that’s my opinion.

Sp. 1: Any last questions or remarks on this interview or at the questions I asked?

Sp. 2: Not really. I think the, I think it is interesting to discuss about it, I think the model itself is rather interesting, it was not clear for me why this model? And not why another model? There are many models that we know.

Sp. 1: I can explain after the interview.

Sp. 2: But then, I would say there’s a little bit of overlap between questions, but, other than that I think it’s quite clear.

Sp. 1: Ok, thank you very much.

Sp. 2: My pleasure.
Sp. 1: Ok, thanks for your time. I would like to start with the first generic question: How important is Continuous Delivery (CD) for your organization and why?
Sp. 2: Well, I think that if I look in the past, we had a lot of focus on functional delivery and we have a lot of big projects arguing over priority. We were starting initiatives and stopping initiatives, which I think was not efficient. That's what I start with. I think the focus lately has been on CD or Continuous Integration initially. But also on test automation to make things faster and higher quality in the first instance. I think a very important thing is if you have engineers who continuously get feedback from users that the software was not meeting the standards. Than you will lose a lot of enthusiasm from the people. Also from the users. And I think that is a big benefit of building quality in your, let's say, delivery pipeline. That the feedback loop would be much more positive and the people will have eventually have more time for, let's say more fun development, than fixing bugs under pressure.
Sp. 1: Ok.
Sp. 2: I think that's the most important outcome. Hopefully.
Sp. 1: Thank you. The second question is: What is your experience in the usage of Continuous Delivery Maturity Models?
Sp. 2: Use of models. I believe zero models. I only know that when we started, that in one of the teams I work with on thinking about this, we were focused on test automation, and we always said that we there's different let's say faces you can have test automation in, unit tests, integration tests, requisition tests, and there we had a little bit of a model, but that was almost purely focused on testing and the tooling around it.
Sp. 1: Ok.
Sp. 2: Yes.
Sp. 1: Yes, clear. What should be in your prospective be the main added value of (using) a CD Maturity Model?
Sp. 2: I think personally that the biggest value of using a model could be the transparency to the team and direct stakeholders. Because it will tell them where we were in the past and as a team/stakeholders. According to our measurement, fine, where are we now? If that is compared, than you see what has been achieved, sometimes the road to get where you want to be is so long, that you forget that you've achieved a lot already. And that will lose momentum. So, in order to keep momentum for the long road, you got to have a vision on where we were and were we are. To be proud of that. Also it can show where we could be. And I think if you involve the participants so that the team members and the direct stakeholders envisioning where we could be or where we wanted to be. That will trigger thoughts on how to get there and the believe that it is achievable. So, that the process of looking at the model, filling it in for your own team, filling in where you want to be and where you were, can have some….. can open up thoughts in people and believe that you can make it. And then off course the question is: what will be our next step then and what is the highest priority so you can set it in motion. So, I think that's the benefit primarily of the model, and
now it is distinguishing between the team and the stakeholders, as a post to comparing teams, and who is the better?

Sp. 1: Yes.
Sp. 2: That’s an alternative potential use. I think it’s far less beneficial.

Sp. 1: Ok, thank you. What are the CD focus areas most relevant for your organizational scope by working Agile/SCRUM in DevOps teams and why?
Sp. 2: So, for my current organizational scope we have a vendor product. And I think a long time in the past the focus has been on other things than automation. And also then I have a little bit of a let’s say culture, since we don’t have access to the servers of the vendor, it’s not up to us to automate. Now, we are moving that around and we are saying, we can’t automate everything, but what can we do? And then the first focus is, let’s do a focus. One is on automating some IT-operations and activities....

Sp. 1: Yes.
Sp. 2: that are recurring, that can be static data, reconciliations, you can do either an Excel may take five hours or eight. So we are focusing on automating these. Makes it more auditable, makes it faster and you can do it more frequently. Meaning that the data will be in line most of the time. So, that’s one monitoring automation is something that we are focusing on, in the past a lot of the team members used to look at the logs, once in a while. And depending on how much time they have available. And now we are focusing on getting alerts let’s say more pro-actively. Only, in case there is an issue. So, that is something that we are focusing on, and in parallel with the vendor, we are focusing on replacing a lot of our manual functional regression tests, with automated tests that run when the vendor pushes code but also when we let’s say snapshot or certain UAT release candidate. And that we are just starting with, but with the vendor we have to believe that we can save a lot of manual regression tests to that way and I think there an important point to mention is that we will save costs, we will save time, but also we will be less dependent on key men who have certain specific knowledge of a functionality that was built five years ago and we keep on.... Let’s say, using them for that purpose. Which is not making them any smarter or having them develop new skills, because they are being used in the same way over and over again. So, that’s about multiple benefits.

Sp. 1: Clear. What critics do you have on CD Maturity Models and why?
Sp. 2: I think, since I’ve not used the models, but I can only imagine that they could be used in the similar fashion for example CMMI, which is also a MM. And where in CMMI is a risk that people start first of all comparing teams and judging based on that. Without knowing the context of that team, without knowing where the team has come from, on what is have been achieved. Just looking at: is it red, orange or green? So that’s a risk.

Sp. 1: There comes the blaming dashboard.
Sp. 2: And there is the blaming dashboard and there is a second risk than, that the teams themselves will feel in order to get a good, let’s say appraisal or feedback from senior management, they need to get things green. And then....
Sp. 1: Yes.
Sp. 2: The goal becomes instead of makes thing green is really improve the whatever you want to achieve. Well, that can leave to at one point of all dashboards being green, but, the real situation been far less green, so, wrong steering information. So, than we missed our goal, and I think that the risk is that at one point, the team therefore will not realize that they themselves can use it. Know they because they were asked by external parties of senior management, they will deliver to senior management and not get the real benefit out of it themselves on how to steer and reflect on their own quality or improvement ideas.

Sp. 1: Ok, clear. Then I have some specific questions on the certain MM I found it’s here in front of you. First one is: How is this MM helpful to determine the status of CD maturity in your organization and why?

Sp. 2: To determine the status of the CD maturity I think it is, there’s always some subjectivity and context involved, so it’s very hard to get a metric on all teams that can be compared. So, but if you want to assess with your team, where you stand, where you were and where you want to be, than I think this can help, because there is an external objective party who doesn’t know your team or doesn’t have a pre-judgement. Who has thought about and interviewed maybe many other companies and who learned and thought about what are keys succes factors? And a model can help you objectively match your organization as you see it, which is a bit subjective, with the model. And then, it can prevent you from overlooking certain important factors. Then I already see one factor in one of the models is culture and organization, which is not something that I think we so far, have...let’s say often associated with CD. But, apparently the researcher of this model has come across other companies or situations, and concluded that it is important. I think it might help you look at the, your organization in a more holistic fashion.

Sp. 1: Good. How is this MM helpful to determine the next steps of implementing CD in your organization and why?

Sp. 2: And then I say a little bit the same as just now, so the access and objective outside authority, so, your team can benchmark itself, and it will also trigger questions like why was this category added to the model? Have we thought about that? How does our team match against this? So, I think it triggers a lot of questions, and reflection on your own team. And if those questions weren’t there and you would start from your own let’s say thoughts, than you would start from your own bias. Which you have perhaps have done the last five years. Which has got you where you are.

Sp. 1: Ok, yes, thank you. What should be extended in the MM and why?
Sp. 2: So I read that the first factor was culture and organization, but when I read through the levels, I did not see, or it was brief, the culture come back a lot.

Sp. 1: Ok.
Sp. 2: So mean, is there, you can’t measure culture clearly, but what are the indicators? No positive indicators, negative indicators or whatever you want to achieve. So I didn’t really see it come back.

Sp. 1: Yes.
Sp. 2: I do think it is a very important aspect in order to achieve this, because people need to buy into it first of all. And, yes, I think it’s very important, so you asked me how can it be extended? Perhaps highlighted further, and also indicate what are potential pitfalls of certain dynamics that can be present in an organization.

Sp. 1: Ok.
Sp. 2: And how do you address those? So, I think a little bit of the people aspect, cause CD often is thought I guess a lot of people think about test automation and that techy thing, but, I
think it has a lot to do if you wanna make it a success with people, so emphasize that more. That’s one of my thoughts.

Sp. 1: Ok.
Sp. 2: What I also think is very important, is we often talk about the fact that we only need a two week planning cycle, so we need to look two weeks ahead and have an emerging design and emerging planning. But, in reality, we see that a big organization needs also have goals for the coming 6 to 12 months or even 18 months. And when those goals are in people’s heads, they wanna reach and achieve them. So, we have to find some way of having operational planning, tactical and strategically planning, live in parallel, maybe with different cycle, focused on by different parties in the organization, senior management might look ahead further as where the scrum team might focused on the coming two weeks, but somewhere they need to sink up.

Sp. 1: Yes.
Sp. 2: And, and if a long term goal comes along, for the coming 6 months, yes, then there should be a forum or a meeting with certain people, in order to imbed that in the organization. And so there’s a conflict I think where on the one hand we feel that we only should plan only two weeks ahead, but at the other hand we know that long term planning or long term vision in goals are important. Now, I didn’t immediately see that come back in the model, I see design and architecture, typically in architecture, when you make an architecture, you do think what do I want to achieve in the coming 6 to 12 months, but that’s technically.

Sp. 1: Yes.
Sp. 2: And that’s not necessarily on priorities and allocation or sinking op multiple teams. So, I think that might be.....

Sp. 1: So that would be a category like scaling or planning and estimating, or....
Sp. 2: I will call it....

Sp. 1: strategic planning, or something like that?
Sp. 2: Yes, so planning and prioritization, horizons and somehow that needs to come together, and I think there, that should be well, you can call it governance or a forum, the people who make those priority calls and the people who give the estimations, they will be rough estimations at first, come together. And what I see now in practice often that’s not necessarily the model, but in practice, is that whenever we talk about 6 months planning, it’s typically a party that is not in de DevOps team who gives the estimates. But, that party will than try to achieve that timeline, but a DevOps team has never given input. So, it’s a conflict that we have right now, a conflict? And leads to a little bit of frustration, leads to a lot of re-planning , and a lot of expectations not being met and if I think if we want to increase peoples happiness and the setting, let’s say meeting the expectations, that someone has set, than we have to think about this somehow. And get a feel on how good we are doing on the area of planning, horizons and prioritization.

Sp. 1: Ok, thank you.
Sp. 2: I think one point as well that I like to add is: what I don’t see yet is, we have multiple parties like product owner, like a DevOps team, a DevOps team is an IT, senior management is an IT and a product owner is in front office. What we often see is that whenever a product owner doesn’t priorities what we find important in IT, then we will force it at one point. He? Like life cycle management and risk issues.

Sp. 1: Yes.
Sp. 2: So, it appears, that a product owner has primarily a different focus. So, I’m thinking, is let’s say goal setting or KPI-setting important? Should we have similar KPI’s? Or the same? And I didn’t see that coming back here, so that’s just a thought. And one other point is, I think
in order to make CD a success, competencies of the people are very important. Engineering skills clearly, but competencies as well, because sinking up multiple scrum teams, really relies on individuals being able to interact, have the assertiveness and the confidence to speak up, share their thoughts, and that’s essential and ideally I think we have as many as possible. Because if we say all the team managers or the integrators should do that, now, you’re relying on 10 or 20 per cent of your people. Where we want to rely on 80 per cent of the people.

Sp. 1: Yes.
Sp. 2: So, I think I don’t see the competencies coming back yet, but in some way I think it’s an important success factor.

Sp. 1: Ok. Question 4 I think you already gave a quite some recommendations, but do you have other recommendations that can be given next to the MM to implement CD efficiently and effectively in your organization and why?
Sp. 2: Yes, I think we should, let’s say ask as many team members and parties involved like product owners, what they see as recurring challenges. Or stressful areas. Openly. So, basically start indeed with gathering inside from the people. Because I have the feeling that often ideas come from management, and those ideas are once course based on some senses that they have had or some things they’ve heard. I think it’s important, to ask open questions to team members and then I think we will get a lot of inside, which will be valuable.

Sp. 1: Ok, thank you.
Sp. 2: And it will engage the people so whenever we implement or make changes to the way we organize ourselves, people will have been part of it and have a bigger chance of buy in.

Sp. 1: Ok. How will you use this maturity model for self-assessment in the future and why or why not?
Sp. 2: Yes, I think again I think the main how I will use it is a and why, the process of using the model is I think a very important one, because people will start reflecting, asking themselves questions, getting stuck, not knowing the answer, talking to each other, I think that process is one of the most important things, whether it than turns out to be green, orange or red, that is the….

Sp. 2: secondary yes. So, I think that’s the most important thing. It engages the team members, and perhaps we will get a lot of ideas out of it in the process.
Sp. 1: Ok, thank you. Why would you recommend this MM to other IT Managers working inside or outside your organization?

Sp. 2: Well, I have to say that when I look at this particular model it contains more than I indeed typically associated with CD. When I typically think of the technical aspects. I think information and reporting is another very important one, that, in our organization it maybe so far under focused on. What you see is that a lot of parties wanna know, how is it going? Do we have impediments, do we have issues, am I getting what I want? When can I test what? And all those questions, lead to e-mails, lead to responses, and lead to more questions, and basically distraction from what we should or want to be doing.

Sp. 1: Yes.

Sp. 2: So, I think and at the same time, yes, so I think looking at this model it's contains a lot more factors than you will not initially think of when you speak of CD. So, I think that's good. And that's why I think it's good to use a model instead of your own thoughts which have bias already.

Sp. 1: Ok, What have you learned from practicing with the CD Maturity model an why?

Sp. 2: I haven't used it yet. I think, no, we only looked at Continuous and yes, a little bit about Continuous integration and testing, test automation, what I've seen is that when we had a workshop for example recently in our team, and when we look openly about CD not necessarily this model, than indeed people started thinking about their own situation in coming up with ideas that..... Hypothetically, they could have had in the past, but, by sitting down making time for it and asking the open question, they answer it with something they've known for years. But, since it wasn't asked, it didn't stated, so not necessarily this model, but by asking how could CD help us? A lot of things came up. And my experience is also that that energizes the people.

Sp. 1: Ok. My last question is number 8: What could be improved in the MM and why?

Sp. 2: I think it's a little bit like what can be what should be extended? Like I mentioned I think people and their competencies are very important.....

Sp. 1: Planning and forecasting you mentioned priority setting......

Sp. 2: Yes, planning and forecasting, KPI's, alignment perhaps, or maybe it's an enabling factor, something that helps. I think highlight more the people and their competencies. Cause often we have I think a tendency to say, let's work harder or let's automate it, and that's not always the solution, some people don't know where to start or they don't know how to make decisions or priority calls or how to let's say flag impediments in a respectful and friendly way to another team or team member. So, we can help them with that. And yes, there is a little bit of risk that we focus a lot on the technical aspects, but we also need to help people along to develop themselves and their fashion. So, yes, is that part of CD or is it an enabler? It's one or the other.

Sp. 1: Ok, thanks a lot. Do you have any other questions or remarks on the questions that I've just asked you?

Sp. 2: No.

Sp. 1: Ok, thank you.

Sp. 2: Thanks.
9.6 Interview 6: IT Manager

<table>
<thead>
<tr>
<th>Interview 6</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of minutes: 12</td>
</tr>
<tr>
<td>Number of speakers: 2</td>
</tr>
<tr>
<td>Language: English</td>
</tr>
</tbody>
</table>

Sp. 1: My first question is: How important is Continuous Delivery (CD) for your organization and why?
Sp. 2: Ok. Yes, it is important as an organization has changed the whole philosophy of IT within the last 12 months, to Agile, Devops, Scrum approach and Continuous Delivery fits with that approach. So, yes, it is very important for the organization, to make this move.

Sp. 1: Ok. What is your experience in the usage of Continuous Delivery Maturity Models?
Sp. 2: None, I have no, I've never seen MM, never heard of it in terms of CD so. For me it's new.

Sp. 1: Ok, question 3 is: What should be the main added value of using a CD Maturity Model?
Sp. 2: A MM gives you an idea on how far you've come, in adapting to a methodology as CD and how far you haven't come, so, how far are you on the journey? And some idea of what gains and advantages can still be achieved?

Sp. 1: Yes, very clear. What are the CD focus areas that are most relevant for your own organizational scope indeed by working Agile/SCRUM in DevOps teams and why? Why are those focus areas most important for your scope?
Sp. 2: Ok, I think our organization. Well, the CD is giving a lot of emphasis put on the deployment.

Sp. 1: Yes.
Sp. 2: The build and deploy. Personally I think a bit too much.

Sp. 1: Yes.
Sp. 2: They, the key benefit and then not just to build and deploy, but the whole different aspects of CD is time to market. I mean and particularly in our organization, time to market is a killer. We are so slow.

Sp. 1: Yes?
Sp. 2: And there are a lot of reasons why we are so slow, but I think the CD the Agile Devops, that's all. In the end you want to be quick. You have to move, be flexible and be quick.

Sp. 1: Ok, yes, very clear. What kind of critics could you have on CD Maturity Models and why?
Sp. 2: Yes, it's rigid maybe, in what level is it a generic thing.

Sp. 1: Yes.
Sp. 2: So, the same, not always specific to your situation, for the DevOps teams that are using it, how to fit them. It's a management thing as well. Can be a management thing to see as doing, that you call it? A control over things how far, judgement, judging teams on, I hate that. With the bottom-up approach about…
Sp. 1: The blame sheet.
Sp. 2: Of agile stuff, you could be careful, we have a bottom-up approach, which I really like about the agile approach. Yes, does it really fit? On the other hand, it does give you an idea of say how far you are. And how possible it is to remain to be uncovered.

Sp. 1: Yes, ok, thanks a lot. Than I have also some specific questions on the CD model of Andreas Rehn: How is this MM helpful to determine the status of CD maturity in your organization and why?

Sp. 2: It’s nicely splits out the variable suspects, so you have to build and deploy, test and validation, but I see other things that might be missed, like culture and organization, design and architecture is also one when I was looking at it this morning, because I read a little bit behind it, I thought: your whole build and deploy what’s often missed, at least have been missed in our company. What does this say about what we have to do about our architecture? I think financial markets in particular have a, we don’t build our own applications, so we’ve bought an application, so that’s very different, but if you look at the architecture, consequently it is basically saying: you need to go into a very modular, not modular but even very small components service type based architecture.

Sp. 1: Yes.
Sp. 2: I’ve not heard people talking about architecture, but they talk about build and deploy.

Sp. 1: Yes.
Sp. 2: So, it’s good. So.

Sp. 1: Brings it in perspective.
Sp. 2: Brings in: hey, we heard about this and have you thought about that?

Sp. 1: Yes.
Sp. 2: So, in terms of the advancement, shows you how far you are coming up.

Sp. 1: Ok, yes, thanks. How is this MM helpful to determine the next steps of implementing CD in your organization and why? Think you said it already?
Sp. 2: I think I covered it much of that he?

Sp. 1: Yes. My question 3 is: What could or should be extended in this MM and why? You miss certain elements that comes to service?
Sp. 2: I’m afraid I haven’t really deep dived into the model.

Sp. 1: What kind of recommendations could be given next to the MM to implement CD efficiently and effectively in your organization and why?
Sp. 2: It’s far broader then just build and delivery or test and verification. And that’s my first: oh yes! When I look at the model, we are broad, this is a broader thing than this. But, that has to be brought back to the teams again, because I think we have been very narrow, with the story of CD and also for our company I think we do not sell, it has not been sold, of the advantages what is meant to bring. I bring the story to the floor and the floor I am in now, is not been well sold.
Sp. 1: So, that should be a really recommendation to a selling method?
Sp. 2: Yes, why what is your advantage doing this? Because, I know people who will think there is no advantage doing this.

Sp. 1: And then you mean from both the IT-side and the Business-side or…? 
Sp. 2: I mean from the IT-side, from the Business-side also I don’t know.

Sp. 1: Ok. Any other recommendations that you’ve seen from your experience that you think like: hey, that’s really important to implement CD in the organization? 
Sp. 2: The aspect around testing is really important. I think there is a bit too much emphasize on the build and deploy, I get really annoyed when people say what to do to 200 deployments at a day or something. Get realistic, but if we can speed up the testing, than the testing is really one of the areas we really are very poor at, and really consumes well. Traditionally it consumes always more than half the time from start to finish. And deployment is fine I think.

Sp. 1: Ok.
Sp. 2: For most teams.

Sp. 1: Thanks a lot. How will you use this maturity model for self-assessment in the future and why or why not?
Sp. 2: The, I, it’s a, I think it could be useful let’s say, for me it’s new; a MM around CD is new. With a sort of checklist for Agile/scrum. But, coming from CD we haven’t anything at all. To my knowledge. So yes, with some stuff in there which is nice just to see how we are doing.

Sp. 1: Ok, thank you. So, maybe a little bit double some questions, but, why would you recommend this MM to other IT Managers working inside or outside your organization? I think you already mentioned like, why there’s nothing, so than it could be recommended.
Sp. 2: So my take is. CMMI also came in, a bit like this this with varies areas, and then what level are you at.

Sp. 1: It’s the blaming sheet?
Sp. 2: Yes.

Sp. 1: That shouldn’t be there. Ok. Did you have learned anything particular from I would say discussing or practicing with the CD Maturity model and why?
Sp. 2: We’ll I’ve not been practicing it, but for what I’ve seen so as I said, the other areas are interesting.

Sp. 1: Ok, thanks. My last question: What could be improved in the MM and why?
Sp. 2: I can’t, it’s too early to say. I’ve not been working with it.

Sp. 1: Ok, thanks a lot.
Interview 7: IT Manager

<table>
<thead>
<tr>
<th>Interview 7</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of minutes:</td>
</tr>
<tr>
<td>Number of speakers:</td>
</tr>
<tr>
<td>Language:</td>
</tr>
</tbody>
</table>

Sp. 1: Thanks for taking the time for this interview, let me start with the first question: How important is Continuous Delivery for your organization, and why is that?
Sp. 2: Ok, I think it’s very important, because the IT-Market is transforming fast, and I think also in particular for the organization, and the area that I work in, quick time-to-market is of crucial importance and it will only get worse. So, the only way to defend ourselves as IT-managers and IT-departments from it, is to automate our own work as well.

Sp. 1: Ok, time to market is really the important thing here?
Sp. 2: Yes, and also the business expectancy, of quickly adapting to different changes. So, through the credit crisis, through IT-innovation in general: people’s expectations also change quite a lot, also in decision making and that also means decision making would changes live and the only thing to do that, is if you have an efficient framework to develop and put things live. So.

Sp. 1: Ok, thanks a lot, it is clear. What is your experience in the usage of Continuous Delivery Maturity Models?
Sp. 2: Zero. So, I heard about Continuous Delivery Maturity Models, I, haven’t worked with them, to be honest, maybe a bit of the next question, one of the next questions as well? It’s a bit….

Sp. 1: Yes, we can go to that question, what should be the main added value of (using) a CD Maturity Model?
Sp. 2: I think it’s the gage where you are, and to have common understanding of where you can go and what would be the next stepping stones. I think, part of the difficulties that we face with implementing continuous delivery in an agile environment is that we leave a lot of things up to the people themselves to realize, so, by doing that, you get a lot of momentum of people trying to implement these things, but, at the same time sometime you have different people lagging in the different let’s say level of expectation. And it helps to define the next step on how to get somewhere. And I think that dialogue is often unstructured, if you leave it fully up to the teams, but we reuse all the best practices known ideas or steps that there are and I think Continuous Delivery Maturity Model is something that could help with that.

Sp. 1: To structure the discussion more as a talking plate or….?
Sp. 2: Yes, so, the bigger vision is always described in textbooks, but it is always what are the connecting dots basically, to achieve that goal, and that’s not always clear to everyone in the team or at least within teams maybe not everyone is at the same level of expectancies. So, you spend a lot of time in defending the next step to try and achieve the final goal. And I think by doing the assessment, you have a little bit of a framework to point you in the right direction and also make you there. I think the pitfall is with the phased Maturity Model, that it becomes some kind of classification as good or bad, because without that, you kill the momentum.
Sp. 1: Yes.
Sp. 2: And the spirit of having pride in what you deliver. Which should be the prime differentiator of exactly wanting to invest in continuous delivery as an employee.

Sp. 1: Ok, thanks a lot. What are the CD focus areas that are most relevant for your organizational scope by working Agile/SCRUM in DevOps teams and why?
Sp. 2: I think the, well for the new department than at least I think the deploy is very handy, because it saves a lot of time and frustration by far the most important at the moment is the testing. Because we lose so much time in testing and retesting, and still there are so many flaws, seeping through and you won’t ever be able to test everything in a perfect way, so, it is a bit also where the theory may go wrong in adding more and more tests, but the ambition is very nice to speed it up and really tackle that. And I think in our organization at least the change department that I have been working in, has been struggling with it since I started working here twelve years ago. And each time in a different flavor try to address the testing and the verification and the responsibility. And a good thing about automating, than the responsibility is clear, it’s the automation and the results should be pretty unambiguous as well. So, I think that would really help also to buy time for people to focus on the real important stuff. Which is in the end adding business value or not. Not testing everything over and over again.

Sp. 1: Ok, thanks, that’s clear. Question 5 is: What critics do you have on CD Maturity Models and why? I think you just already mentioned a critic, it is not being seen like some kind of management dashboard and killing momentum. And the other critics, on the Maturity Continuous Delivery Maturity Models?
Sp. 2: Not really, but, that’s also because I haven’t had that much experience with them already so, I couldn’t really say. Just, indeed it shouldn’t kill the momentum. And it shouldn’t serve as a classification of being right or wrong because than you lose the objective of doing something Agile/SCRUM.

Sp. 1: Ok, really clear. Maybe for look on the CD maturity model of Rehn: How would this MM be helpful to determine the status of CD maturity in your organization and why?
Sp. 2: I think what you can see on it is that it gives the stepping stones for each of the different let’s say aspects of continuous delivery, so, it should be easy to measure a team or an application or environment on where they are and then, to look at where the most opportunity for the next step is. And I think you may have a bit of added value or state in any of the different disciplines so if you are very good at deployment, that doesn’t make you necessarily that good at testing it’s good that it gives an overview of the full picture.

Sp. 1: Ok. And how is this MM helpful to determine the next steps of implementing CD in your organization?
Sp. 2: I think that’s what we have to gage separately is on where we would want to implement this indeed. I think the next step is visible just in the diagrams, so it’s more the question of putting it through the tests.

Sp. 1: And what you just said, by valuing what is really in it for the organization?
Sp. 2: Exactly.

Sp. 1: Yes.
Sp. 2: So, Yes.
Sp. 1: That also is already an input for question 3: What should be extended in the MM and why? There is one thing that is not in the MM not really the value, so there is really something that teams or organizations should make up themselves do you see other things that should or could be extended in the MM?

Sp. 2: Yes, it’s the use on the prime thing that sense out. I would say. No, not really without going to a lot of details for specific applications because you can consider things like tooling and how efficient use tooling or consolidation level, across applications, for example.

Sp. 1: Ok.
Sp. 2: But, that’s the thing with the theory it doesn’t describe it which level you want to apply this. So, maybe the authors and the inventors did think of it as being applicable to different levels in the organization, but it is something that I cannot read from it at the moment.

Sp. 1: The tooling aspect I’ve heard in interviews before already also is an important one. Ok, What recommendations can be given next to the MM to implement CD efficiently and effectively in your organization and why?
Sp. 2: I think it’s indeed try to address the business value and convince the business from participating in it. Because, without it, you get a lot of end discussions like we’re having today on the value of investing in test sets.

Sp. 1: Good idea. Not now.
Sp. 2: Not now. So, it’s more, somehow we have to find the time to do, to keep getting our own house in order. And I think that’s something that we should keep on fighting for. And by the way, I haven’t seen it any different in other banks or organizations that I worked for, or even in terms of doing consultancy in the telecom sector. We had the same kind of discussions. So, for me it is not something new... for the bank.

Sp. 1: Ok. How will you use this maturity model for self-assessment in the future and why not?
Sp. 2: I think that would be a very good exercise actually you know, we have action on it, with the CD road map. It’s something that in your day to day function you’ve asked for as well. That we are struggling with in our department. Also has to do with having people see what’s missing. I think it would be a good exercise to just have them draw this up and then say: is this our next level and also to celebrate the success. So in zero touch deployment indeed very near and hoera it’s all working, so, we’re champions of that and be able to share it. So, I definitely would suggest to experiment with it.
Sp. 1: Ok.
Sp. 2: If we do, I would also do it on... it would be interesting to see it at the department level or let's say, on the feature level as well. So, to take for example different applications and really string up the different pieces of the chain together and then to see how it differs. So, is there a difference between how you asses a team or how a teamwork application assets itself versus if you add it all up.

Sp. 1: Do you take the minimum? Or do you then have the pipeline or is it exactly subject to other kind of....
Sp. 2: Yes, that’s a good one.

Sp. 1: I think it’s a good idea to put it around that bar.

Sp. 2: Ok, that means also that you’re.... Why would you recommend this MM to other IT Managers working inside or outside your organization?
Sp. 1: Wow, one thing is because it addresses one of the things that one our teams are struggling with at the moment, so, it probably could push them from very theoretical discussion on the end side. Or, very practical, let’s say the only small step implement the security rights in, but actually get the next step on what’s then in scope and I think, that focus could really help also them making that step, and making some visible progress around it, so yes.

Sp. 1: Ok, clear. Question 7: What have you learned from practicing with this CD Maturity model and why?
Sp. 2: I don’t have an answer for you, because I didn’t apply the Maturity Model yet.

Sp. 1: Yes, ok. And last, I think you already talked about some improvements but I ask the question: What can be improved in the MM and why?
Sp. 2: I think value, because it is the most important selling point to the business to actually keep on implementing this and it’s also something that will come back, so it will be very nice instead of saying you achieve the benefits on the end, and to already say yes. Step one, you already achieve this....

Sp. 1: Tooling you....?
Sp. 2: Yes, and that in lead of tooling as well, so what kind of tooling or ideas could you offer for each of the different phases to prevent people from stepping right away to the expert level. So, I may be totally wrong, because I don’t know the content of this specific one, but if you take a tool like Nolio right. Is it now an expert level thing, that you only have to bring in at the end, or is it actually something that you have to do in step one? And for version control it is pretty clear, it’s also one of the first things, so what are then the common market practices just to make it more tangible for people if they want to start to do a next step on what will be the next level of tooling you think about? Otherwise, people re-invent the wheel all the time and that’s just not the smart thing to do.

Sp. 1: Ok, any other questions or remarks?
Sp. 2: No, just really interested to see the thesis....

Sp. 1: It’s coming. Thanks a lot.