

Nordea

Software Process introduction

January 2017



Comments regarding the training material to EDC

- This “self-study-package” is a high level view of the Software Processes in Nordea
- The material is especially targeted to the EDC consultants in India
- The aim is primarily to raise the awareness of the Software Processes in Nordea
- To fulfil the need of a “self-study-package”, each slide is accompanied with notes to support the studying

Content

- Scope of the software processes
- Software process models
 - Agile software process
 - Traditional software process
- 9 Domains
- Software development lifecycle in a project
 - Decision points
 - Templates
- Quality checks and gates
- Summary



The aim is to acquire general and overall knowledge about Software Processes in Nordea.

Focus is more on what to deliver rather than how to do.

More detailed information can be found from the Software processes –portal in Step In via Nordea Intranet (called – ‘Intra’).

There is also list of mandatory deliverables and links to templates which are needed.

Scope of the software processes

"Mature Processes Improving Quality!"

- The Software Processes guides projects and teams in adhering to Nordea standard and guidelines in each of the 9 software delivery domains.
- In this context software process mean software development life cycle and software process model across 9 domains.
- The Software Processes, apply to software and software development in the broadest possible sense: from traditional, transaction based mainframe software via Internet-of-Things Apps to Cloud and virtualisation software.
- All *IT related* projects, initiatives and maintenance efforts, *incl. infrastructural ones*, are subject to the standards and guidelines described in the Software Processes.
- Software life cycle management standards governing in Nordea applies to both developmental activities and maintenance.



4

Nordea

The 9 domains in the previous slide (circle) and on this slide will be described in details later in this material. Some domains are detailed in separate presentations and some in the Nordea intra pages by their owner organisation (e.g. Change Management).

Software process models

- Nordea is moving gradually toward agile way of working. However there is still lot of initiatives which are using more traditional approach and currently many new initiatives still select traditional process for their way to work.
- Some initiatives use kind of 'hybrid model' which is mostly agile, but contains some traditional steps. Many initiatives has taken full scale agile approach and following Scaled Agile Framework (SAFe).
- According to Nordea project model initiatives shall be classified either traditional or agile. Following slides elaborate these models, Agile and Traditional, more thoroughly.

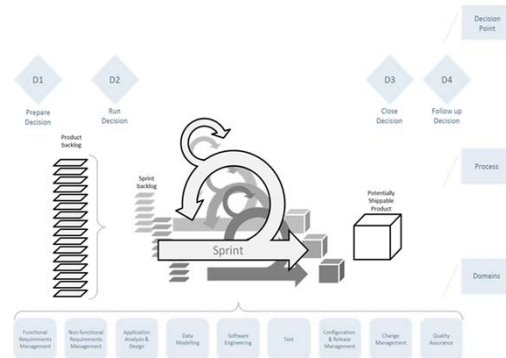
The aim is to introduce variety of the development models used in Nordea.

Software process models

Agile Software Process

- Agile Software Process aims at early delivery of customer value by frequent delivery of small portions of usable functionality and is compliant with the Agile Manifesto and the 12 principles behind it.
- The **9 Domains** included in the Agile Software Processes are defined to ensure a clear definition of interaction in activities, tasks, responsibilities and expertise, resulting in higher customer satisfaction and faster software delivery.

Agile Software Process is compliant with the principles in the internationally known Agile manifesto (key concepts: Individuals and Interaction, Working Software, Customer Collaboration and Responding to Change) and is compatible with any Agile framework e.g. Continuous Integration, Safe, Scrum, Kanban etc.



Agile Software Process

• Key concepts in Nordea's Agile Software Process

- The **Product Owner** represents all relevant stakeholders and has discretion over content and priority of any item on the Product Backlog.
- The **Product Backlog** is a list of (prioritized) functional and non-functional requirements usually referred to as epics, features or user stories in ascending order with items that meet the demands in the **Definition of Ready (DoR)** when they are used for software development and moved to the **Sprint Backlog**.
- The **Sprint Backlog** is the list of user stories to be delivered in the current sprint with items that meet the demands in the **Definition of Done (DoD)**.
- During a sprint the **Agile Development Team** conducts analysis, design, development, testing and deployment in a steady cadence of once every 2 – 4 weeks.
- An **Agile Development Team** is a multi-disciplinary team that together processes all skills and experience of the 9 development domains to deliver working software.
- **Frequent delivery**: The Agile Development Team delivers working software (referred to as **Potentially Shippable Product - PSP**), meeting the demands of the **Release Definition of Done (RDoD)**, when handed over to Operations and Maintenance.

The Product Owner and Agile Development Team are jointly responsible for the quality of the delivered software as well as delivering mandatory governance deliverables, such as Product Vision, Master Test Plan, Solution Architecture Design (SAD), etc., and adhering to the demands in the IT Quality Checklist and Spot checks.

- A business case and Early Architecture Assessment (EAA) is created as input for decision point D1.
- An initial Product Backlog is created as input for decision point D2, but can and will be extended afterwards.
- Only software that meets the demands of the Release Definition of Done is put into production.

Scrum term 'Sprint' is used here as an example. Initiative might have also iterations or cycles.

Definition of Ready and other criteria items are described later in this material.

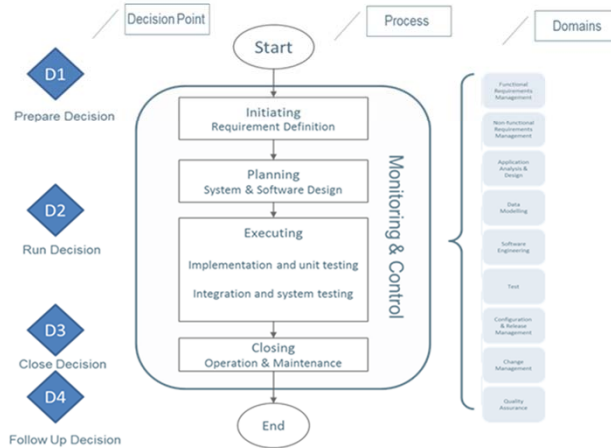
Decision points D1 to D4 are also described later.

Development models

Traditional Software Processes

- **The Traditional Software Process**, on both Waterfall and Iterative, is compliant with the international standard ISO/IEEE 12207, covering all processes concerned with full life cycle of software products including maintenance.

The Traditional Software Process is flow based and runs through a series of stages that relies on up-front planning and a set of sequential, prescribed steps to e.g. eliminate all project risks. Each stage typically has its own team of experts and scripted milestones. The Iterative approach being used in Nordea is following the same sequential steps, where actually the activities between Decision Point 2 and 3 are iterations.



Traditional Software Processes

- **Key concepts in the Traditional Software Process**
 - **The Sponsor:** The overall accountable of the Project/ Programme.
 - **Programme Manager:** Accountable to the Sponsor for the successful delivery of the Project outcome.
 - **Project Manager:** Accountable to the Sponsor or to the Programme Manager for the successful delivery of the Project outcome.
- The process is connected to the Decision Points used in the Nordea Project Governance Approach. Each Decision Point is a checkpoint where it is decided if the requirements are met in order to move on to the next stage in the Process.
- The 9 Domains included in the Traditional Software Processes are defined to ensure a clear separation of activities, tasks, responsibilities and knowledge, resulting in better planning and management.

Traditional Software Processes

Planning

- Functional and non-functional requirement specifications are studied.
- System and Software Design are prepared. System Design helps specifying hardware and system requirements, to define overall system architecture. Activities for the project or change are planned during this stage, resulting in a detailed planning. Also the Master Test Plan or Level Test Plan will be created during this stage.
- Requirements are translated into an overall system architecture. Software design involves representing the software system functions in a form that can be transformed in to a executable application/solution. The Design phase is best described by breaking it up into Logical Design and Physical Design sub phases. During the Logical Design phase, system's analysts makes use of the information collected in the Requirements phase to design the system independently of any hardware or software system. Once the higher-level Logical Design is complete, system's analyst begins transforming it into a Physical Design dependent on the specifications of specific hardware and software technologies.

Traditional Software Processes

Executing

- The goal is to implement the software solution into the production environment. All activities, that needed to get to this point, are part of the Executing stage.
- Each individual component has to be developed and tested to ensure that each unit functions properly with respect to its specification before the units are integrated. Then integrate the individual components and test the system as a whole to ensure that entire software system functions properly with respect to its specifications and environment.
 - Implementation and unit testing – During this stage the software design is realised as a set of program or program units. Unit testing involves verifying that each unit meets the design, specification and functional and non-functional requirements.
 - Integration and System Testing – The individual Program units or programs are integrated and tested as complete system to ensure that the software functional and non-functional requirements have been met. After acceptance testing, software system is delivered to production for integration in production environment.
- Once all the functional and non-functional testing is done, product is deployed in the production environment.

Traditional Software Processes

Closing

- Purpose is to close down the development initiative and handover to Operations and Maintenance.
- The system is delivered and integrated in production and the customer is using the developed application. After a handover period where the project team is still responsible for solving defects and issues, the solution is officially handed over to Operations.

Iterative Process

An iteration encompasses the development activities that lead to a product **release** - a stable, executable version of the product, together with any other peripheral elements necessary to use this release. Therefore, development iteration is in some sense one complete pass through all the disciplines: Requirements, User Interface Design, Analysis & Design, Implementation, and Test, at least. It is like a small Waterfall project in itself.

9 Domains

The Software processes divides solution development in 9 distinct domains, that together cover all aspects of solution development. For core activities, this is a repetitive cycle. Support activities have interfaces with the surrounding environment or exists in all areas (Quality Assurance).

Core activities are covered by these 6 domains:

- Functional Requirements
- Non-functional Requirements
- Application Analysis and Design
- Software Engineering
- Data Modelling and
- Test

The support activities by these 3 domains:

- Change management
- Configuration and Release Management
- Quality Assurance

Domain is an area that covers one activity or stage in the Solution development that requires specific knowledge and/or expertise and that produces specific, well defined deliverables

9 Domains

Functional Requirements Management

- Functional Requirements Management is the process of eliciting, analysing, refining, and prioritizing **functional** product requirements and planning for their delivery. These activities are also known as Requirements Development.
- Milestone deliverables for Functional Requirements Management are:
 - Product Vision
 - High Level Functional Requirements
 - Detailed Functional Requirements

Non-functional Requirements Management

- A Non-Functional Requirement (NFR) is a requirement that considers attributes such as Usability, Reliability, Performance, and Supportability. NFR's make out the "URPS+" in FURPS+ (Usability, Reliability, Performance and Supportability, other like Resilience and Security).
- Milestone deliverables for Non-Functional Requirements Management are:
 - High Level Non-Functional Requirements
 - Detailed Non-functional Requirements

Non-functional Requirements Management has also own material to describe it more thoroughly.

9 Domains

Application Analysis & Design

- The Application Analysis and Design domain addresses all static, dynamic and behavioural modelling aspects of Nordea's application landscape. It builds upon the requirements (both functional and non-functional). Where requirements primarily focus on **what** the system shall do, AA&D focuses on **how** it will be implemented.
- Milestone deliverables for Application Analysis & Design are:
 - Architecture Vision
 - SAD - Solution Architecture Document
 - Business Process Diagram
 - Use Cases
 - User Stories

Data Modelling

- Data Modelling addresses Nordea's logical data entities and their interrelationships. It describes **which** data is used in a particular area or sub-area.
- Milestone deliverables for Data Modelling are:
 - Data model
 - Data Transportation Language
 - Data categories

9 Domains

Software Engineering

- The Software Engineering domain addresses the core software delivery tools and practices such as designing, coding, code review, unit testing, building, integrating and deploying code artefacts.
- Milestone deliverables for Software Engineering are:
 - Software artefacts
 - UI Design
 - Build and Deployment instructions

Test

- The Nordea Test Strategy sets the framework for all test activities within Nordea. The purpose of the Nordea Test Strategy is to achieve uniform, standardized, efficient, traceable and high quality test in Nordea. Everybody working with IT changes must adhere to and follow The Nordea Test Strategy.
- Milestone deliverables for Test are:
 - Master Test Plan / Level Test Plan
 - Master Test Report / Level Test Report
 - Quality Assurance and Production Gate

Software Engineering domain is currently decentralised in Nordea. Each Business IT areas have defined their own methods for it.

Test domain has also own material to describe it more thoroughly.

9 Domains

Configuration and Release Management

- In Nordea Configuration and Release Management is divided into two separate area in the IT Processes: Service Asset & Configuration Management and Release & Deployment Management.
- Milestone deliverables for Configuration & Release Management are:
 - Configuration Management Plan
 - Service document
 - Release Notes

Change Management

- In Nordea Change Management is the process of planning and coordinating the implementation of IT changes.
- Milestone deliverables for Change Management are:
 - Change Management Plan
 - Change Request

IT Processes are maintained by Infrastructure and Operations unit they have their own portal to describe them.

9 Domains

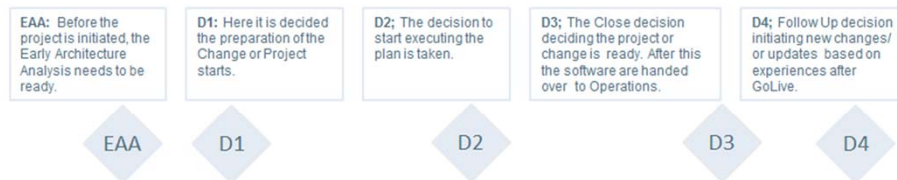
Quality Assurance

- The Quality Assurance domain addresses the framework for Quality aspects during the lifetime of the software development projects. The Quality decision is an addition to the Project Governance based on the approval of the IT Quality Checklist, DoR, DoD, RDoD and Production Gates.
- Milestone deliverables for Quality Assurance are:
 - IT Quality Checklist
 - DoR, DoD and RDoD
 - Production Gate

Quality Assurance domain is also decentralized. Each Business IT areas have defined their own methods for it.

Software development lifecycle in a project

Software Delivery follows Nordea's project governance structure of which the decision points are depicted below. In each decision point certain criteria shall be fulfilled before project can start next phase. Main part of the software processes are applied between decision points D2 and D3.



The table in the next page contains the *abstract*, high level deliverables that are used in the Decision Points.

Some of the abstract deliverables are domain specific, most of them are not. The abstract deliverables are mentioned in the domain that has the lead in creating the tangible deliverable as well as in the domains that play an important, recognisable role in doing so.

Project lifecycle is described more in Project and Portfolio Management portal (PPM) in Nordea Intranet or in delivered Project Management material.

Theme of the high level deliverables that are used in the Decision Points

	EAA	D1	D2	Executing the project	D3	D4
Functional Requirements Management	Early Architecture Assessment	Business Case	Product Vision High Level Functional Requirements	Detailed Functional Requirements		
Non-functional Requirements Management			Product Vision High Level Non-functional Requirements	Detailed Non-functional Requirements		
Application Analysis & Design			Software Architecture	Updated Software Architecture		
Data Modelling			Software Architecture	Updated Software Architecture		
Software Engineering			Software Architecture	Software Artefacts UI Design Deployment Instructions		
Test			Master Test Plan	Level Test Plan Anomalies	Test report	
Configuration & Release Management			Configuration Management Plan	Base Lined Packages Release Notes Configuration Item(s)		
IT Change Management			CAB approved IT Change	Change Request(s)		
Quality Assurance			Quality Criteria	Quality Criteria	Met Quality Criteria	Closure Document

Deliveries from each domain are not limited on these shown in the table. Naming might be also different.

Main purpose of this table is to show type of deliveries expected in each decision point. Some deliverables are mandatory to all developed products and some are conditional, depending scope of the development.

Nordea Domains Templates

Nordea templates are based on best practices and methodologies and follow the industry standards. These templates are selected to cover the entire life cycle on software development process for both Traditional and Agile.

Domains	Templates	Remarks
Functional Requirements Management	<i>Product Vision, Use Case Diagram, Use Case Model Survey, Use Case Specification, Software Requirements Specification (SRS)</i>	All Requirement templates can be found from Software Process portal
Non-functional Requirements Management	<i>Product Vision, Specification Non-Functional Requirements (Word format), Specification Non-Functional Requirements</i>	All Requirement templates can be found from Software Process portal
Application Analysis & Design	<i>Solution Architecture Document (SAD), Solution Design Document (SDD), UI Design templates</i>	Application Analysis & Design is supported by the MEGA toolset.
Data Modeling	<i>Solution Design Document (SDD)</i>	Data Modelling is supported by the MEGA toolset.
Software Engineering	<i>Solution Architecture Document (SAD)</i>	
Test Management	<i>Mandatory Metrics, Level Test Plan, Level Test Report, Master Test Plan, Master Test Report</i>	All Test templates can be found from Test Portal
Configuration and Release Management	In Remedy and in IT Process portal	Templates for Service Configuration Management Templates are implemented in Remedy. For templates for Release Management please see IT Process portal.
Change Management	Change Templates in Remedy	The existing change templates are available via the change module in Remedy.
Quality Assurance	Embedded in Clarity	The IT Quality Checklist template is embedded in Clarity and can be found in Clarity/Properties module.

21

Nordea

MEGA – toolset used for data modelling and Application analyses & Design.

Remedy – tool used for issue and change management.

Clarity – tool to track, control and report progress of the projects.

Software Process portal – Nordea internal portal for software processes.

Test portal – Nordea internal portal for testing methods and steps.

IT Process portal – Nordea internal portal for the processes maintained by Infrastructure and Operation unit.

Quality Checks & Gates

Acceptance / Quality criteria

- Following this framework is mandatory for:
 - IT Projects
 - IT Improvement Initiatives
 - IT Maintenance (as applicable)
- For a software change or project in Nordea it is mandatory to follow the general prerequisites described in **IT Quality Checklist** as much as applicable.
- Initiative specific criteria shall be described in own documents.

Current situation is that Agile projects doesn't have IT Quality Checklist in Clarity. Agile projects can use Initiative Prerequisites and Release Definition of Done (RDoD) instead until IT Quality checklist is available for Agile.

	Project	Improvement
Traditional	Scope Statement: acceptance criteria	Improvement description: acceptance criteria
Agile	Project Description: Definition of Ready (DoR), Definition of Done (DoD) and Release Definition of Done (RDoD)	Team Description: DoR, DoD, RDoD

Required documents and content for both model and scale.

IT Improvement Initiative – small development initiative which doesn't have to use project governance model.

IT Quality Checklist, Initiative Prerequisites, Definition of Ready, Definition of Done and Release Definition of Done are described in next pages.

Acceptance criteria for traditional development shall be defined by per initiative in to Scope Statement or Improvement description. There is no specific template for it. Agile development usually have acceptance criteria embedded in to User Stories and common criteria is described in DoR, DoD and RDoD.

Quality Checks & Gates

Gates

- For all software initiatives in Nordea it is mandatory to pass all the Decision Points defined in the Project Management Framework.
- Quality criteria before GoLive decision is defined in: **Production Gate (Agile/Traditional)**

IT Quality Checklists

- The purpose of the IT Quality Checklist is to prevent incidents in production caused by release of changes. It supports the project in ensuring that the quality of the solution delivered will be on sufficient level.
- The Project Manager is responsible of the IT Quality Checklist and will be supported by the Project Management Team when assessing the project according to the checklist. Responsible IT Partner is responsible for approving the IT Quality Checklist.
- The IT Quality Checklists define *when* during a project lifetime certain criteria must be fulfilled. It is recommended to use as a supporting tool during the project life cycle not at least when preparing and planning the Project / the next phase in the Project. It is possible to mark most of the criteria N/A or Partly as well as Yes and No.
- The IT Quality Checklist is embedded in the Clarity tool.

Initiative Prerequisites

- When the work is initiated it is important for successful delivery of sustainable software that the way is paved for the Agile team.
- If the initiative is organised as a project, it is the Project Manager who is responsible of the Initiative Prerequisites and the Responsible IT Partner who is accountable and approves the Initiative Prerequisites are in place.
- If the initiative is organised as a team (feature team / improvement team), it is the Team Lead who is responsible of the Initiative Prerequisites and the Responsible IT Partner who is accountable and approves the Initiative Prerequisites are in place.
- The Initiative Prerequisites do together with the Definition of Ready and Definition of Done per product backlog item and release secure sustainable quality (operational stability, robust code, non-functional aspects, sufficient test etc.).
- It is mandatory to fulfil the Initiative Prerequisites though separate documentation of the fulfilment is not required.

Currently each initiative shall define their own Initiative Prerequisites. Some examples can be found from Software Process portal.

Definition of Ready, DoR

- Definition of Ready is a list of expectations that refined business desirements, stored in the Product backlog as backlog items, must live up to in order to be ready for system development.
- Definition of Ready is focused on the correctness of the work results and securing sustainable quality of the system developed (operational stability, robust code, non-functional aspects, sufficient test etc.).
- Backlog items should also have acceptance criteria. The nature of acceptance criteria is concerned with acceptance of the delivered functionality (that should provide value for the customer).
- It is the Agile team who is responsible of the backlog items to be developed are ready according to the Definition of Ready and the Product Owner is accountable and approves that the backlog items are ready. It is mandatory to fulfil the definitions though it is not required to document the fulfilment per backlog item, the responsibility and accountability is placed on specific roles.

Each initiative shall define their own Definition of Ready. Some examples can be found from Software Process portal.

Definition of Done, DoD

- Definition of Done is a list of expectations that software increments (deliveries of backlog items) must live up to in order to be released into production.
- Definition of Done are focused on the correctness of the work results and securing sustainable quality of the system developed (operational stability, robust code, non-functional aspects, sufficient test etc.).
- It is the Agile team who is responsible of the backlog items are done according to the Definition of Done and the Product Owner is accountable and approves that the backlog items are done at the same time as it is secured that the acceptance criteria for the backlog item are fulfilled. It is mandatory to fulfil the definitions though it is not required to document the fulfilment per backlog item, the responsibility and accountability is placed on specific roles.

Each initiative shall define their own Definition of Done. Some examples can be found from Software Process portal.

Release Definition of Done, RDoD

- Release Definition of Done is a list of expectations that a software release must live up to in order to be released into production.
- Release Definition of Done are focused on the correctness of the work results and securing sustainable quality of the system developed (operational stability, robust code, non-functional aspects, sufficient test etc.).
- It is the Agile team who is responsible of the release are done and ready for production according to the Release Definition of Done and the Product Owner is accountable and approves that the increments in the release are done and can be released to production.
- It is mandatory to fulfil the definitions though it is not required to document the fulfilment of the definitions, the responsibility and accountability is placed on specific roles.

Each initiative shall define their own Release Definition of Done. Some examples can be found from Software Process portal.

Production Gate

- The main purpose of the Production Gate is to ensure that software and hardware changes to be implemented will not have a negative effect on the production environment.
By checking all mandatory steps on testing and on the software processes before the GoLive decision, it is checked if mandatory documents and processes have been followed up.

Risk Assessment Questions and Regular Spot Checks

- In the Regular Spot Checks conducted on Projects Improvements or Maintenance areas it is validated that the test related risk question issues are documented in accordance with the Test Strategy 3.0.

Improving IT operational stability

- The strict change freeze was decided in order to obtain IT operational stability. The main purpose of the Production Gate is to support stability after the strict Change freeze is lifted. Also more data of risk and impact of changes are collected to be used to better target future improvements.

Summary

Main items in a high level to be remembered from this presentation:

- Two alternative process model for the development; Agile and Traditional
- 9 domains across development models with their core and supporting activities.
- Project lifecycle and decision points
- Deliverables for the decision points and domains to contribute.
- Templates for the domain deliverables.
- Quality checks and gates to ensure quality and compliance.

Nordea

Thank you!

