Focused Build for SAP Solution Manager 7.2 (SP5)
Requirement, Work Package and Work Item

Customer Experience & Solutions, SAP SE
December 2019

PUBLIC
Disclaimer

This presentation outlines our general product direction and should not be relied on in making a purchase decision. This presentation is not subject to your license agreement or any other agreement with SAP. SAP has no obligation to pursue any course of business outlined in this presentation or to develop or release any functionality mentioned in this presentation. This presentation and SAP's strategy and possible future developments are subject to change and may be changed by SAP at any time for any reason without notice. This document is provided without a warranty of any kind, either express or implied, including but not limited to, the implied warranties of merchantability, fitness for a particular purpose, or non-infringement. SAP assumes no responsibility for errors or omissions in this document, except if such damages were caused by SAP intentionally or grossly negligent.
Content

Change Paces in Requirement to Deploy

Requirement Planning: Example SAP Activate

Manage Requirements

Manage Work Packages

Manage Work Items
Change Paces in Requirement to Deploy
SAP Solution Manager Integration Model
Three Different Change Paces in Requirement to Deploy

<table>
<thead>
<tr>
<th>Fix</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Trigger</td>
<td>Business disruption or standard change</td>
</tr>
<tr>
<td>Scope</td>
<td>Small</td>
</tr>
<tr>
<td>Process Impact</td>
<td>None</td>
</tr>
<tr>
<td>Approval</td>
<td>Individual</td>
</tr>
<tr>
<td>Deployment</td>
<td>Unbundled on request or bundled with release</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Enhance</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Trigger</td>
<td>Improvement request</td>
</tr>
<tr>
<td>Scope</td>
<td>Medium</td>
</tr>
<tr>
<td>Process Impact</td>
<td>Minimal</td>
</tr>
<tr>
<td>Approval</td>
<td>Individual</td>
</tr>
<tr>
<td>Deployment</td>
<td>Bundled with release</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Innovate</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Trigger</td>
<td>Transformation projects, new solutions</td>
</tr>
<tr>
<td>Scope</td>
<td>Large</td>
</tr>
<tr>
<td>Process Impact</td>
<td>Significant</td>
</tr>
<tr>
<td>Approval</td>
<td>Pre-Approved</td>
</tr>
<tr>
<td>Deployment</td>
<td>Bundled with release</td>
</tr>
</tbody>
</table>
SAP Solution Manager Integration Model

Process Flow

The Requirement-to-deploy value chain supports the three different change paces at an optimum.

<table>
<thead>
<tr>
<th>Fix</th>
<th>Enhance</th>
<th>Innovate</th>
</tr>
</thead>
<tbody>
<tr>
<td>Program fix required to resolve disruption</td>
<td>Enhancement required for daily business operations</td>
<td>Strategic initiative for new business model</td>
</tr>
<tr>
<td>Fix immediately, deliver break-fixes and standard changes</td>
<td>Assess enhancement request, negotiate delivery and cost</td>
<td>Model to-be processes, collect Requirements</td>
</tr>
<tr>
<td>As fast as needed.</td>
<td>Deliver enhancement</td>
<td>Plan solution delivery</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Deliver solution with continuous business feedback</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Bundled in major release</td>
</tr>
</tbody>
</table>

No overhead.

Monitor Solution Readiness

---

The Requirement-to-deploy value chain supports the three different change paces at an optimum.
SAP Solution Manager Integration Model
Focus Build Document Flow

<table>
<thead>
<tr>
<th>Fix</th>
<th>Enhance</th>
<th>Innovate</th>
<th>Deploy</th>
</tr>
</thead>
<tbody>
<tr>
<td>Incident</td>
<td>Business Requirement</td>
<td>Requirement</td>
<td>Solution Readiness Dashboard *</td>
</tr>
<tr>
<td>FB Request for Change</td>
<td>IT Requirement</td>
<td>Work Package, Scope Change</td>
<td></td>
</tr>
<tr>
<td>FB Urgent Change</td>
<td>Change Document</td>
<td>Work Item</td>
<td></td>
</tr>
<tr>
<td>FB Standard Change</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Immediately after approval</td>
<td>Minor releases</td>
<td>Major releases</td>
<td></td>
</tr>
</tbody>
</table>

Solution Readiness Dashboard *
SAP Solution Manager Integration Model

Transaction Types

- **Fix**
  - Incident → S1CR → S1HF / S1SG → Immediately after approval

- **Enhance**
  - SMBR → SMIR → SMHF / SMMJ / SMGC / SMAD → Minor releases

- **Innovate**
  - S1BR → S1IT / S1IR → S1MJ / S1CG → Major releases

Solution Readiness Dashboard *

*) Planned innovation: Merge of business requirement and requirement, Solution Readiness Dashboard across the lifecycle
Initial Requirement Planning: Example SAP Activate mapped on Focused Build
Solution Readiness Dashboard
Tracking of Requirement Status based on Live Data - Example

Tile Requirements
Gives an overview of all Requirements assigned to your Master Project
Each bar of the tile is clickable and offers a filtered list of Requirements

Unassigned Requirements
The bar Unassigned Requirements displays all Requirements where still Work Packages needs to created for
How to come to an Initial Backlog - Summary

Steps

Step 1: Requirements are gathered with the help of **Fit-Gap Workshops**
- The creation of an Initial Backlog is a process which typically lasts several weeks or month
- The Requirements definition process can vary very much, dependent of the SAP Partner and Implementation methodology

Step 2: Working with processes and **process structures and diagrams**
- Best Practice is, to start with predefined SAP Content (Best Practice or Model Company) and directly maintain the Requirements in SAP Solution Manager. The advantage is, that process structure elements are automatically assigned and all documents are automatically stored at the correct place
- This data is then inherited by each follow-up document, e.g. the Work Packages and Items
- Working with diagrams coming with the SAP Content and prepared by the SAP consultant beforehand the workshop accelerate the discussion
- Alternatively it’s as well possible draw processes on brown paper and gather the Requirements in xls. When this is done you can bring the Requirements with one upload in SAP Solution Manager. Then the process structures in Solution Documentation need to be maintained, optionally the process diagrams redrawn, and after the Requirement upload, the process structure manually assigned to the Requirements.

Step 3: There is a slim **approval workflow** for the Requirements
- During the Fit-Gap Workshops the Requirements are in Status ‘Draft’
- When the Gap definition for a scenario or process is ready, the responsible Business Analyst sets the Requirement on status ‘To be Approved’

Step 4: The **approval procedure** for Requirements for each release is done by an **Approval Board**
- This is commonly done via the Mass Change Analysis, where the ‘To Be Approved’ Requirements are collected and checked
- Requirements which are not so urgent are put on status ‘Postponed’ and are re-checked at the next Approval Workshop for the next Release
- The activity shall be executed by an Architect (having change authorization for the Mass Change) but under the participation of the program and project managers
Release Planning: Roles and Responsibilities

1. Define Product Backlog
2. Prioritize Product Backlog
3. Analysis of Dependencies
4. Estimate Product Backlog
5. What would you like in the release?
6. Scope Release: Build logical and technical packages

High-Level Release Plan Based on Master Project

Detailed Release Plan based on Build Project
Build Design: Example New Implementation

Fit Gap Analysis and Delta Design – Methodology Overview

**Step A**

Fit Gap Analysis / Solution Validation

1. Finalize System Setup
   - Prepare additional sample data
   - Enhance system setup with add. Scope

2. Fit Gap Workshops / Validation of SAP Solution
   - Show and tell SAP standard key design elements / Gap Identification

3. Gap Documentation
   - Document and specify identified Gaps in initial Backlog

4. Delta Scope Prioritization
   - Prioritize
   - Request the release
   - Should be part of release with Mass Change app

5. Delta Design
   - Slice the Requirements into Work Packages which need to fit in one Wave
   - Document relevant configuration
   - Document solution for identified gaps e.g. One Delta Design for Sourcing and Procurement

6. Verify & Accept
   - Verify solution design
   - Acceptance Procedure

**Step B**

Delta Design

- Create solution design
- Acceptance Procedure

Workshops per Line of Business, e.g. Sourcing and Procurement:

- Present solution based on delta design
- Capture open issues if required
- Obtain sign off

Approve Requirements to be part of release, with Mass Change app

Repeat steps 5 & 6 at the beginning of each Wave

Result: Product Backlog

- Create Work Packages to be part of the first Wave with Requirement Management app
- Repeat this Wave planning activity for each Wave -> agile
- Plan all Waves at once -> waterfall
- Result: Product Backlog

Central review of identified gaps:
- Completeness
- Prioritization

Rework Requirements via Requirement Management or My Requirements app

Capture and classify identify gaps in backlog:
- Document Requirements
- Classification (process gap, functional gap, etc)
- Initial solution proposal
- Initial effort estimates

Extend Processes which are not part of Best Practices, e.g. QM

Best Practices / Model Companies uploaded to SAP Solution Manager

Fiori vs. GUI

Initial Backlog

<table>
<thead>
<tr>
<th>Product Backlog</th>
<th>Prioritize</th>
<th>Order</th>
<th>Should</th>
<th>Must</th>
<th>Could</th>
<th>Would</th>
</tr>
</thead>
<tbody>
<tr>
<td>Priority / SL</td>
<td>16</td>
<td>4</td>
<td>15</td>
<td>5</td>
<td>14</td>
<td>1</td>
</tr>
<tr>
<td>Days</td>
<td>1</td>
<td>5</td>
<td>12</td>
<td>7</td>
<td>11</td>
<td>8</td>
</tr>
</tbody>
</table>

Create Work Packages to be part of the first Wave with Requirement Management app

For real Gaps and WRICEF, create Functional Specification in My Work Packages app

© 2019 SAP SE or an SAP affiliate company. All rights reserved. PUBLIC

14
Once all activities/outcomes of the Fit/Gap Workshops are called ‘Gap’ we distinguish in Focused Build a bit more and ‘classify’ Requirements, Work Packages and Work Items in the following way:

**Gap**: is a completely new development which needs to be specified in detail with big As-is, To-be evaluation, often with no technical information or idea how to realize it in the beginning. Technical design is fully done by developer in WI with the help of a technical design document.

**WRICEF**: is a typical and from SAP expected extension, where no business background needs to be described. The consultant often already knows how to implement and configure. So the Specification is often already a mixture between functional and technical design.

**Fit**: there is no coding adjustment, but only customizing. So specification is often an existing standard configuration guide and only a customizing documentation is needed to be maintained on Work Item level.

**Non-functional**: Is used for documentation upload or a parameter settings without the need to document a Functional Specification (no document KPI maintained in customizing for Work Package and Work Item).
Agile Project Execution with Focused Build

- **Prepare**
  - Jump Start
  - Maintain Requirements
  - High Level Release Planning (Master Project)

- **Explore**
  - Pre-built system or pre-assembled solution
  - Baseline Build
  - Solution Validation
  - FIT/GAP WS Analyst

- **Realize**
  - High Level Release Planning (Build Project)
  - Detailed Release Planning (Build Project)

- **Build**
  - Wave 1
    - Sprints 1-3
    - Delta Design
    - Detailed Planning
  - Wave 2
    - Sprints 4-6
    - Delta Design
    - Detailed Planning

- **Support**
  - Development
  - Test
  - Demo/Test

- **Deploy**
  - Realize Build
  - Realize Plan

- **Run**
  - Realize Plan
  - Realize Build
Manage Requirements
Transparent Requirements-to-Deploy
Agile Delivery Model

**Project Team Onsite**
- Process and Application Landscape
- Requirements
- Work Packages / Work Items
- Solution Readiness Dashboard
- Perceived Gaps
- Fit
- Onsite Delivery

**SAP**
- ICC
- MCC
- Gap validation
- SAP Standard Development

**System Integrators**
- WRICEF/Gap
- Build Factories
- Development Factories
- JIRA Integration
Digital Business
Use SAP Activate for SAP S/4HANA for delta scoping

SAP Activate provides SAP S/4HANA best-practice process content that consists of process diagrams, documentation, and configuration.

- You can download this content into your SAP Solution Manager
- From the diagram, you can jump into SAP FIORI apps in a pre-activated SAP S/4HANA trial system
- Execute show and tell of the SAP S/4HANA innovations hands-on
- Document Requirements as a result of this fit/gap analysis
What’s in Focused Build on top of SAP Solution Manager?

Explore Phase

Requirements Management

- Create Requirements in Validation workshops with customers for delta identification (Fit/Gap analysis)
- Requirements are integrated into process context allowing better handover to build team
- Multi-language support for Requirements
- Tight integration of Requirements Mgmt. into solution landscape, process models, Work Packages, Work Items, Solution Documentation, and the Solution Readiness Dashboard
Best Practice is to add Requirements to process steps and processes, but they can also be linked to other elements if required.

Requirements can be maintained at the following SolDoc Elements:

- Structure Elements
  - Process
  - Process Step (Reference)

- Library Elements
  - Configuration Item
  - Process Step (Original)
  - Executable
  - Development
  - Interface
Document Requirements
Display of Requirements in Process Management

Requirements for Process Steps
Relationship from Requirements to processes in Solution documentation

Feature details
• Indicate in the process if Requirements are available
  • 4 different decorator icon shows the different status of Requirements
  • Preview of Requirement in pop up window
• Direct access to all existing Requirements for the process step to create further Requirements

Benefits
Monitor in the process structure how many Requirements exist and which status
Avoid creating redundant Requirements for the same process
How to start - Methods for Requirement Maintenance or Upload

Overview

Process Management Diagram
➔ Directly maintain Requirement at Business Processes or Business Process Steps

Process Management Column Browser
➔ Directly maintain Requirement at Library or process structure elements

Excel upload
➔ Initial upload of Requirements into Requirements Management. Manual assignment to Process Structure as follow-up activity

Excel down-/upload
➔ Initial down- and upload of process structures including Requirements, e.g. from cloud or Model Company systems
Down and Upload Requirements (since SP03)

Use Case 1 - Other SAP Solution Manager system as data source
Down and upload Requirements with Solution Documentation structure information, e.g. process, process step, library

Use Case 2 - External data source
Upload without Solution Documentation structure information, e.g. process, process step, library.
Use Case 1: Down and Upload Requirements from other SolMan

Applications
Requirements Mgmt., Excel

Feature
• Down and Upload requirements from other systems like Model Company or CALM

Use Case
The project preparation phase started on another SolMan system, like a CALM or Model company.

There process structure and Requirements already have been defined.

Now you can do an initial upload of the Requirements to the design branch of the on-premise SolMan with automated assignment to the process structure.
Use Case 2: Upload Requirements from External Data Source

Applications
Excel, Requirements Mgmt.

Feature
- Upload requirements from other data sources

Use Case
The project preparation phase without another SolMan system, e.g. xls-based or gathered in documents
You can do an initial upload of the Requirements to Requirements Management app. of your SAP Solution Manager
Create the process structure/diagrams and assign them to the Requirements.
SP05 Requirements Management Process Flow
Detailed Workflow

- **Draft**: Send for Approval → Approve
- **To Be Approved**: Approve → Approved
- **Approved**: In Realization → Realized
- **Realized**: Completed (FINI)
- **Rejected**: Set by WP → To Be Approved
- **Canceled (FINI)**: Canceled → To Be Approved
- **Postponed**: Postpone → To Be Approved

- **Send for Approval**: manual status setting
- **Approve**: automated status setting
- **Withdraw**: manual status setting
- **Postpone**: manual status setting
- **Recover**: manual status setting

© 2019 SAP SE or an SAP affiliate company. All rights reserved. | PUBLIC
Key Takeaways Requirements Management

- Are created on the Design branch and not released to Development branch
- Should be related to any process structure in SAP Solution Manager process management, process or non-process related
- Should be specific and granular
- Have an 1:m or n:1 relationship to Work Packages
- Are ideally consolidated (not x Requirements for the same need)
- Need to be approved
- Get an automated status update from the Work Package
- Maintain categories reflecting your build projects
**Document Maintenance with dropDoc**

**dropDoc**

- **Definition:** dropDoc helps as a part of Solution Documentation to manage numerous file types, which simplifies the default usability of file management inside Solution Documentation. dropDoc can be integrated as a part of Work Package (WP), Work Item (WI) and Requirement applications. In addition, dropDoc can be implemented directly, as a standalone option in the Solution Documentation scenario.

**An extract of dropDoc features**

- insert files using **drag and drop for processes and steps**
- **mass maintenance** of documents and documents type
- change the **document status**
- delete one or more documents at the same time
- optimized for different screen resolutions
- Etc.
Basic functionality and usage overview

**dropDoc Attachments**
- in Work Package and Work Item Applications
  - UI5 technology
  - Mass upload of documents
  - Drag and Drop functionality
  - Access to document templates

**dropDoc Attachments**
- integration to Requirement Application
  - Integration into Requirement Application
  - Mass upload of documents into selected Requirement
  - Direct access to document templates
  - Drag and Drop functionality
Basic functionality and usage overview

**dropDoc Attachments in Work Package and Work Item Applications**

**dropDoc Attachments integration to Requirement Application**
Functionality of dropDoc

dropDoc integration to Work Package and Work Item Applications

STANDARDS

SAP Solution Manager allows to define document standards, by central definition of document types, their templates, allowed usages and completeness rules.

The list is providing the minimum documentation standards and their typical usage.
Functionality of dropDoc

dropDoc integration to Work Package and Work Item Applications

STANDARDS

- Process description
- Use case
- User guide (training material)
- Functional integration test case description
- Single functional test case description
- Functional specification
- Functional specification for enhancements
- Functional interface specification
- Technical specification
- Configuration guide
What’s New with Focused Build SP5?

Requirement Management
Requirements Management - New with SP05

Requirement Upload Report

Applications

Requirements Management, Upload report
/SALM/Requirement_Import

Features

• Additional options how to handle the Requirement Owner field
  • Overwrite existing with current logon user
  • Keep original Business Partner
  • Check and take-over in case Business Partner exists in source and target system
  • Complete Requirements with current logon user in case the Business Partner is not maintained

• Extended consistency checks
  • Completeness of obligatory fields
  • Duplicate Requirements
  • Success of Process Structure mapping (creation of standalone Requirement in case of

Use Case

The extended checks ensure better data quality and less clean-up effort after the upload
My Requirements, My Work Packages, My Work Items - New with SP05

Extended Change History

Applications
My Requirements, My Work Packages, My Work Items and the related CRM Web UI

Features
The change history has been extended by the following values

- Requirement
  - Value Points, Effort Points, Local Flag
- Work Package
  - Value Points, Effort Points, PPM Project, Wave, Dates
- Work Item
  - Value Points, Story Points, Dates

Use Case
E.g. in My Requirements app several fields are open for changes after approval. As it is compliance relevant, a Change History is needed to see, if and who changed the values at a later point in time.
Product Backlog Planning: Example SAP Activate mapped on Focused Build
Build Design: Example New Implementation

Fit Gap Analysis and Delta Design – Methodology Overview

**Step A**
Fit Gap Analysis / Solution Validation

1. Finalize System Setup
   - Prepare additional sample data
   - Enhance system setup with add. scope

2. Fit Gap Workshops / Validation of SAP Solution
   - Show and tell SAP standard key design elements / Gap Identification

3. Gap Documentation
   - Document and specify identified Gaps in initial Backlog

4. Delta Scope Prioritization
   - Prioritize according to criteria (e.g. Business Value, Criticality)

5. Delta Design
   - For real Gaps and WRICEF, create Functional Specification in My Work Packages app

6. Verify & Accept
   - Verify solution design
   - Acceptance Procedure

**Step B**
Delta Design

5.1 Create Product Backlog
   - Prioritization according to technical criteria (e.g. Development sequence)

5.2 Design/Specify
   - Specify the work to be done, e.g.
     - Create Delta Design Documents
     - Upload Configuration Documents
     - Prepare distribution of work by definition of Work Items

6. Verify & Accept
   - Verify solution design
   - Acceptance Procedure

Repeat steps 5 & 6 at the beginning of each Wave

Workshops per Line of Business, e.g. Sourcing and Procurement:
- Present solution based on delta design
- Capture open issues if required
- Obtain sign off

To sign-off release Functional Specification in My Work Packages app and set according Work Package Status

Create Work Packages to be part of the first Wave with Requirement Management app

Repeat this Wave planning activity for each Wave -> agile

Plan all Waves at once -> waterfall

Result: Product Backlog
Terminology  Mapping SAP Activate – Focused Build

- **Final Build**
- **Release**
- **Epics**

**Initial Backlog**
- **Content**
  - **Fit / Gap analysis**
  - WS
  - Gap
  - Prioritized/ Ranked, rough effort estimation
  - Fit
  - Requirements
  - Gap, WRICEF, Fit (only configuration)
  - Non functional Requirements

**Product Backlog**
- **Work Packages**
  - Gap, WRICEF, Fit (config) Non functional Requirements
  - Functional analysis & release Planning (aggregation or splitting) duplicates eliminated

**Time**
- **Release**
- **Waves**
- **Sprints**
- **Sprint planning** (Which dev team, which order)

**User Stories**
- **Work Item**
- **Work Item**
- **Work Item**
How to come to a Product Backlog and create Work Packages - Summary

Steps

The Creation of Work Packages is not an activity an Architect executes in one attempt, but in a phased approach:

Step 1: When the Requirements are approved for the first release, the Architects create the Work Packages
  - Create WPs for each Build Project, e.g. Purchase to Pay, Logistics, Controlling, Master Data, …
  - Doing that, they build the Product Backlog for the first or all Waves. So a WP should be developed in one Wave.
  - Planning for the first Wave, only, means a more agile attempt as you are fully flexible for the next wave. Planning for all waves would be typical for a classical waterfall approach

Step 2: When the Product Backlog Planning for the first Wave is done, the Functional Specifications or Configuration Guides are created. Ideally, Test Cases or at least the Templates are assigned as well.

Step 3: When the planned Development is clearly described in the Functional Specifications, the needed development activity can be described and distributed. This is done by scoping Work Items for the Developers

Step 4: Once the Planning activities are finalized, the Architect approves the Scope -> as a result the Work Package is set to status Scope finalized.
  - This activity can be done by an Architect in his single Work Packages
  - Or in form of an Approval Board Workshop with the help of the Mass Change Analysis

Step 5: When the Development starts all Work Packages are handed over to development
  - Like above this can be done by each Architect or commonly via the Mass Change Analysis
Release Planning: Roles and Responsibilities

1. Define Product Backlog
2. Prioritize Product Backlog
3. Analysis of Dependencies
4. Estimate Product Backlog
5. What would you like in the release?
6. Scope Release: Build logical and technical packages

Business Responsibility
- Business Analyst

IT Responsibility
- Architect

Create Requirement based on business process
Use Value points, effort and priority to rank
Approve, postpone Requirements

High-Level Release Plan Based on Master Project
Detailed Release Plan based on Build Project
Work Package and Work Item Classification

- In case a WP has a specific Classification
- This has an influence on the potential (Sub-) Classification of a Work Item (see table below)

<table>
<thead>
<tr>
<th>WP Classification</th>
<th>Gap</th>
<th>WRICEF</th>
<th>Fit</th>
<th>Non-Functional</th>
</tr>
</thead>
<tbody>
<tr>
<td>WI Classification</td>
<td>Gap</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Workflow</td>
<td>Workflow</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Report</td>
<td>Report</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Interface</td>
<td>Interface</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Conversion</td>
<td>Conversion</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Enhancement</td>
<td>Enhancement</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Form</td>
<td>Form</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Fit</td>
<td>Fit</td>
<td>Fit</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Non-functional</td>
<td>Non-functional</td>
<td>Non-functional</td>
<td>Non-functional</td>
<td></td>
</tr>
</tbody>
</table>
Agile Project Execution with Focused Build

Prepare

Realize

Build

Explore

Sprints

1-3

Delta Design

Wave Planning

Highlevel Release Planning (Master Project)

Initial Backlog

Requirements

Value Effort

10 4

6 5

23 1

3 8

14 2

9 7

34 3

10 4

39 2

44 2

63 4

44 3

57 3

57 3

98 4

96 5

Detailed Release Planning (Build Project)

Product Backlog

Work Packages

Value Effort

Would

10 4

20 4

6 5

23 1

14 2

34 3

63 4

44 3

52 3

89 6

98 4

96 5

Could

20 4

6 5

23 1

14 2

34 3

63 4

44 3

52 3

89 6

98 4

96 5

Should

Must

Wave 1

Wave 2

Demo/Test

Sprints

4-6

Wave Planning

Detailed Release Planning (Build Project)

Product Backlog

Work Packages

Value Effort

Would

10 4

20 4

6 5

23 1

14 2

34 3

63 4

44 3

52 3

89 6

98 4

96 5

Could

10 4

20 4

6 5

23 1

14 2

34 3

63 4

44 3

52 3

89 6

98 4

96 5

Should

Must

Support
Manage Changes with Work Packages
Create Work Package
Guidelines

- Bundling Requirements and Work Packages (WPs) follow the organizational structure of the team, e.g. Purchase to Pay, Logistics, Controlling, Master Data, …
- Each team works on its Product Backlog which is the summary of the assigned Work Packages

- Work Packages are linked to a process structure or library element
  - To automatically inherit the Process Documentation context, create WPs based on Requirements
  - Exception can be “basis module” customizing: then there is no direct link to a Requirement, but a direct link to the process structure, e.g. Configuration Item in the Configuration Library

- Work Package Title
  - Reuse Requirement Title (in case of 1:1), e.g. SO_via_Rigo_fixed_handling_fee
  - Reuse Requirement Title plus additional specification of needed deliverable (in case of 1:m), e.g. SO via Rigo fixed handling fee – Logistics interface
  - In case multiple Requirement are defined and summarized in one WP (in case of n:1), define a suitable Work Package title

- A Work Package has at least one Functional Specification assigned
- A Work Package needs to be testable by an end- or key user (functional test) in one wave

- A Work Package can be subdivided into one or more Work Items in case the work needs to be distributed between several project members
Create Work Package
Guidelines

A Work Package needs to be implementable in one wave (~4-12 weeks)
-> the smaller the Wave duration, the better the tracking and status based reporting

Example Waterfall approach with fix planned waves - rough Wave planning according to the WP type
• Wave 1: Fit (Customizing) -> basis configuration
• Wave 2: Gap (Development) -> bigger developments, with integrative aspects
• Wave 3: WRICEF -> smaller developments and adjustments
• Wave 4: Left overs and Integration Testing

Example Scaled Agile approach with fix planned waves
• Wave 1: Plan WPs with the priority according to the ranking in the Product Backlog. Create appropriate Specification
• Wave 2: Plan WPs with the priority according to the ranking in the Product Backlog. Extend appropriate Specification
Wave 3: like above

In case a Work Package is too big to finalize it in one Wave, slice it in several and smaller parts, e.g.
• one WP for developing the basic functionality
• one WP to extend the functionality.
• one WP for finish the functionality
Create Work Package
Guidelines

Rule of thumb: you need to find a balance between detailed planning/description and efficiency/flexibility.

➔ Enable functioning of standard business process with high level business process description as functional specification
➔ Focus on gaps and fits for the process during next wave. Here you focus on the details and specific extension to standard transactions or own developments
Focused Build methodology – Branches

Standard Workflow

Design Branch

- Approved
- In Realization

Development Branch

- Created
- Scoping
- Scope Finalized
- To Be Developed
- In Development
- Single Functional Test & AT
  - In Repair
  - To Be Tested
  - Successfully Tested

Production Branch

- Realized
- Completed

Requirements

- Requirements

Work Package

- Work Package

Work Item

- Work Item

= Automated action
= Manual action
= WI with Normal Change
= WI with General Change
= WI with General Change

In Repair

In Development

To Be Tested

Successfully Tested

Handed over to Release

In Realization

To Be Developed

In Development

Successfully Tested

Handed over to Release

Productive/Completed

Created

In Development

To Be Tested

Successfully Tested

Handed over to Release

Productive/Completed

© 2019 SAP SE or an SAP affiliate company. All rights reserved. | PUBLIC
Focused Build methodology – Systems
Standard Workflow

SBX

- Approved
- In Realization

Create Work Package

- Created
- Scoping
- Scope Finalized
- To Be Developed

Create Work Item

QAS

- In Repair
- To Be Tested
- Successfully Tested

Single Functional Test & AT

- Handed over to Release

FIT & RT

PRE

- Realized

PRD

- Completed

- Productive/
- Completed

DEV

- Created
- In Development
- To Be Tested
- Successfully Tested

Unit Test

- Hand over to Release

Create Transports

- Created
- In Development

Create Work Item

- Created
- Scoping
- Scope Finalized
- To Be Developed

Create Work Package

- Approved
- In Realization
Result

Work Packages are created and documentation is assigned

- In Focus Build we limit the documentation to the really required documentation
- So there needs to be a document describing what needs to be customized and tested
- The availability of the functional specs can be tracked via the Solution Readiness Dashboard
Functionality of dropDoc

dropDoc integration to Work Package and Work Item Applications

CREATE AND ASSIGN DOCUMENTS

There are several possibilities available to create a document in dropDoc.

The document is always automatically assigned to the structure and consequently to WP/WI.
Functionality of dropDoc
dropDoc integration to Work Package and Work Item Applications

CREATE AND ASSIGN DOCUMENTS

1. Create documents by drag-and-drop
   - several documents can be selected from the local storage and dropped over the drop area
Functionality of dropDoc

dropDoc integration to Work Package and Work Item Applications

DEFINe DOCUMENT

The decision where the dropped document will be stored is done in the background by dropDoc and based on the standard document type configuration e.g.:

- **Functional specification** shall be stored at <Step origin>
- **Single Functional Test** at <Step origin>
- **Technical Design** at development or executable elements
- **Use Case** at <Step reference>

<table>
<thead>
<tr>
<th>Description</th>
<th>Document Type</th>
<th>Element Type</th>
<th>Status</th>
</tr>
</thead>
<tbody>
<tr>
<td>ZFC Post goods rece...</td>
<td>Functional Specification</td>
<td>Mass maintenance</td>
<td>Not started yet</td>
</tr>
<tr>
<td>ZFT Post GR.docx</td>
<td>Single Functional Test</td>
<td>Mass maintenance</td>
<td>Not started yet</td>
</tr>
<tr>
<td>ZTD Goods receipt.docx</td>
<td>Technical Design</td>
<td>Mass maintenance</td>
<td>Not started yet</td>
</tr>
<tr>
<td>ZUC Post GR.docx</td>
<td>Use Case</td>
<td>Mass maintenance</td>
<td>Not started yet</td>
</tr>
</tbody>
</table>

Next Close
Focused Build methodology – Document Upload

Standard Workflow

- Approved
- In Realization

Requirements

- Created
- Scoping
- Scope Finalized
- To Be Developed
- In Development

Work Package

- Created
- In Development

Upload Func. Spec (Spec. Status = Released)
Create Work Item type NC for Transports incl. Func. Spec. + Tech Design
Create Work Item type GC for Test Cases

Single Functional Test & AT

- In Repair
- To Be Tested
- Successfully Tested

- Hand over to Release
- Productive/Completed

Unit Test

- Created
- In Development
- To Be Tested
- Successfully Tested

- Hand over to Release
- Productive/Completed

© 2019 SAP SE or an SAP affiliate company. All rights reserved. | PUBLIC
Direct Creation of Work Items in WP Status To be Tested

In My Work Packages you can create Work Items in Work Package status ‘To be Tested’

**Benefit**

Early Single Functional Test: To upload Test Cases in Status ‘To be Tested’ you can now directly create Work Items. So there’s no need to go in back in status Scope Extension to create the Work Item there.
What’s New with Focused Build SP5?

Work Package / Work Item
Requirement Management, My Work Packages - New with SP05
Create Work Package - Solution Filter for Project Display

Applications
Requirement Management, My Work Packages

Features
Creating new Work Packages, you need to choose a project. Now the Project selection is limited to Projects part of the selected Solution

Use Cases
Use Case 1:
In case in Requirements Management there is a filter set for one Solution and you select one or more Requirements from the result list -> in Project selection pop-up, only projects assigned to this one Solution are displayed

Use Case 2:
In case of standalone WP -> in Project selection pop-up, all projects of all Solutions are shown

Use Case 3:
In case there is no Solution filter selected and there are Requirements of more than 1 Solution selected -> pop-up informing, that it is not allowed to create Work Packages of different Solutions
My Work Items - New with SP05
Withdraw Work Item – Improved Handling of Solution Documentation

Applications
My Work Items

Features
Work Items cannot be withdrawn if they have Solution Documentation Elements assigned.

Use Cases
So far, all Solution Documentation Elements have been activated to Production Branch if a Work Item was withdrawn. Now the system checks for changed elements, if there is at least one the system blocks the withdrawal. The changed elements must be either discarded or moved to another Work Item.
My Work Items - New with SP05
Improved Transport Handling in Work Items

Applications
My Work Items

Features
- Long-running Import of Transport of Copies does not block the status set to “To be tested”
- Work Item cannot be set to ‘Successfully tested’ if a transport task was added directly in Development System while Work Item is in status ‘To be tested’

Use Cases
When a Transport Request is quite huge it can happen that the release and import of the Transport of Copy last longer than the system checks for the successful import. So far the status was then reset to ‘In Development’, although the import was successful. Now the system allows to set the next status with an error-flag that can be rechecked afterwards if the transport was successful.

When developer create a task directly in the Development System while the status of the Work Item is ‘To be tested’, they cannot confirm now the successful test with the release of the Transport Request. Instead the status is set back to ‘In Development’ to ensure that the last version of the Transport Request with all included tasks is unit-tested.
Manage Changes with Work Items
Focused Build methodology – Document Upload

Standard Workflow

Requirements

1. Approved
2. In Realization

Work Package

3. Created
4. Scoping
5. Scope Finalized
6. To Be Developed
7. In Development

Work Item

8. Created
9. In Development

Single Functional Test & AT

10. In Repair
11. To Be Tested
12. Successfully Tested
13. Handed over to Release

FIT & RT

14. Realized
15. Completed

Upload Test Cases (via Work Item)

- Upload Tech Design to WI type NC (Spec. Status = Released)

- Upload Test Cases (Test Case Status = Released)

- Hand over to Release

- Productive/Completed

- Completed

- In Development

- Handover to release

- Productive
Relation between Solution Documentation, Requirements, Work Packages/Items

Overview

- **Process & Wave**
  - Requirements
    - Requirements Description
  - Work Package
    - Functional Specification
    - Test Case
    - Interface Specification
  - Work Item
    - Technical Specification
    - Configuration Guide
  - Sprint
    - Draft
    - Approved
    - Completed
    - Productive
    - Scoping
    - Postponed
    - Rejected
    - Created
    - Successfully tested
    - To be tested
    - To be developed
    - Successfully tested
  - Configuration Library
    - To be tested
    - Successfully tested
    - Created
    - Handed over to release
    - To Be developed
    - Postponed
    - Rejected
    - Completed
    - Productive
    - Created
    - Successfully tested
    - To be tested
    - To be developed
    - Successfully tested
  
- **Processes**
  - Process
    - Process Step Library
    - Executable Library
    - Development Library
    - Interface Library
  
- **Library**
  - Interface Library
  - Configuration Library
  - Development Library
  - Executable Library
  - Process Step Library

© 2019 SAP SE or an SAP affiliate company. All rights reserved. | PUBLIC
Tasks for Work Items

It’s possible to create tasks for Work Items
• There is a new “Tasks” tab in the “My Work Item” application to create/edit tasks
• For flexible use, the simple Task status schema is decoupled from Work Item status

Benefit

Break down of work to be done for a Work Item into smaller chunks to help developers organizing their work.
Manage Changes with Defect Corrections
(for more details see L2 presentation Defect & Defect Correction)
Integrated Defect Correction

Best Practice is to use the Integrated Defect Correction, which means:

- Automatic assignment of Defect Corrections to Work Package with visibility in ‘Scope’ Tab
- New Work Package Status ‘In Repair’
- In WP Status ‘Handed over to Release’ correlated Defect Corrections are switched to this Status

Benefit

While Single Functional Test: Existing Defect Corrections are automatically assigned to a Work Package in case of Assignment Analysis Usage and a 1:1 relationship of WP and Test Package.

Automated switch of Work Package Status in case of new and confirmed Defect Correction
Go-Live
(for more details see L2 presentation Deployment and Release)
What happens with Work package, Work Item and Requirement at Go-Live
Combined Productive and Completed status

- When switching to release phase ‘Hypercare’ the Work Packages and Work Items must be in status ‘Completed’ to express they were finished in the current release. The status ‘Productive’ is not sufficient.

- It is not possible to set the status ‘Hypercare’ in the Release if there are non-completed Work Packages and Work Items with status ‘Handed over to Release’ or later (e.g. ‘Productive’, ‘Preliminary Import Requested’)

- Default since ST-OST SP2: Batch Import automatically sets (via asterisk setting, triggering standard after import status setting) Status ‘Completed’ for *Normal* Changes (automated change with transport)

- To have an additional status ‘Completed’ with the FINI status instead of the status ‘Productive’ doesn’t seem offer additional value. But the benefit of this behavior is: Enabling the 4 eyes principle for *General* Changes (manual change without transport) in productive environment
Focus Build methodology
Standard Workflow at Go-Live

Parallel documentation activation to Production Branch and transport to Production System for Work Items (NC)

Work Items (GC) are set via Mass Change to ‘Productive’ which triggers the activation of Solution Documentation Elements to Production Branch

Handed over to Release cannot be set for Work Packages if assigned Documents and Test Steps are not released

Warning appears at “Successfully tested” in Work Items to inform developer about missed activities
Thank you.