How to Perform Backlog Planning for Agile Projects with Focused Build

Henrik Zimmermann, SAP
December 05, 2019
Legal disclaimer

The information in this presentation is confidential and proprietary to SAP and may not be disclosed without the permission of SAP. This presentation is not subject to your license agreement or any other service or subscription agreement with SAP. SAP has no obligation to pursue any course of business outlined in this document or any related presentation, or to develop or release any functionality mentioned therein. This document, or any related presentation, and SAP’s strategy and possible future developments, products, and platforms, directions, and functionality are all subject to change and may be changed by SAP at any time for any reason without notice. The information in this document is not a commitment, promise, or legal obligation to deliver any material, code, or functionality. 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 noninfringement. This document is for informational purposes and may not be incorporated into a contract. SAP assumes no responsibility for errors or omissions in this document, except if such damages were caused by SAP’s willful misconduct or gross negligence.

All forward-looking statements are subject to various risks and uncertainties that could cause actual results to differ materially from expectations. Readers are cautioned not to place undue reliance on these forward-looking statements, which speak only as of their dates, and they should not be relied upon in making purchasing decisions.

For all recent and planned innovations, potential data protection and privacy features include simplified deletion of personal data, reporting of personal data to an identified data subject, restricted access to personal data, masking of personal data, read access logging to special categories of personal data, change logging of personal data, and consent management mechanisms.
Prepare Phase - Project Preparation
Initial Backlog
Definition

An Initial Backlog in Focused Build means to have:

- A list of approved and prioritized Requirements
- Building the content of the to be implemented Release

A Release can be assigned to a:

- Project -> more Waterfall approach
- Wave -> more Agile approach

There must be an aligned and conscious decision how agile the project should be implemented and organized. This is less a question of implementation tools, but how much a team should/is able to change it’s way of work.
Transparent Requirements-to-Deploy
Release Management to synchronize projects and keep them under control

Reduce Risks, Simplify Go-Live Process, and Decrease Test Efforts
Release Assignment in Project

Prerequisite

A Release can be assigned to a Project...

... or Wave
Fit Gap Analysis and Delta Design – Step 1
Prepare a Fit/Gap Workshop

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

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

**FB Role:** Business Analyst

**Functionality:** Process Management

**Procedure:**
- Choose a Sandbox or On-Premise SAP Solution Manager System
- Check predefined content offerings from SAP (SAP Best Practice or Model Company content)
- Choose and further prepare the business process(es) to be part of the Fit/Gap analysis
- In case it is in the knowledge of the consultant or there are already more documents or process models available, extend and adjust the predefined process diagrams

**Advantage:**
- Accelerated set-up of Sandbox system
Use SAP Process Content
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.
Create Work Package
Activate Best Practices with Focused Build

Step 1
Baseline build, as basis for Requirements creation in the Fit/Gap Workshops, shall be done on the Sandbox system:
- By default, it is not foreseen to maintain Requirements and Work Packages in Discover and Prepare project phase
- So, just activate the Best Practices via Solution Builder without explicit Focused Build specific reporting

Step 2
For the configuration of the Development system, it’s recommended to create one Requirement per Business Process. There are several options dependent of the configuration complexity and duration.
- Simple and quick configuration done in one Wave: 1 Requirement, 1 WP, 1 WI, 1 Transport
- Simple but long lasting config. done in several Waves: 1 Requirement, X WPs , 1 WI, 1 Transport
- Complex but quick config. done in one Wave: 1 Requirement,1 WP , X WIs, X Transports
- Complex and long lasting config. done in several Waves: 1 Requirement, X WPs , X WIs, X Transports

The Work Packages which specify the Solution Builder activities are classified as FIT.
Explore Phase - Plan Initial Backlog
Release Planning: Roles and Responsibilities

1. Define Product Backlog
   - Business Analyst, Project Manager

2. Prioritize Product Backlog
   - Use Value points, effort and priority to rank
   - Approve, postpone Requirements

3. What would you like to have in the Release?
   - Initial Backlog = High-level Release Plan based on Master Project

4. Analysis of Dependencies

5. Estimate Product Backlog
   - Create Work Packages

6. Scope Release: Build logical and technical packages
   - Product Backlog = Detailed Release Plan based on Build Project and Wave

Business Responsibility:
- Business Analyst
- Project Manager

IT Responsibility:
- Architect
Fit Gap Analysis and Delta Design – Step 2
Execute a Fit/Gap Workshop

Step 2: How to execute a Fit / Gap Workshop?

FB Role: Business Analyst
Functionality: Process Management
Naming: Gap in Activate is called Requirement in Focused Build

Procedure:
- Logon to SAP Solution Manager and review the business process (either in Process Structure or Diagram)
- Maintain Requirements (= Gaps in Activate)
  - Document Requirements in SAP Solution Manager as draft in Process View or Diagram. Requirements in FB are classified as Gap, WRICEF, Fit, Non-functional).
  - Alternatively it is possible to document Requirements in xls or another system e.g. from Model Company, and upload them to On-premise SAP Solution Manager system later.
- Release Process to Development Branch (with the help of a Work Package and Work Item).

Reporting:
- Solution Readiness Dashboard (Unassigned Requirements) /Requirements Management

Advantage:
- Efficient Fit-Gap Workshops: Requirements are linked/displayed to Business Process enabling an efficient discussion and reducing the duration of the workshops
Document Requirements
Example: Maintain Requirement at a Process Step

Requirements can be assigned within the diagram editor or in the browser by selecting the process step or other structure elements.
Requirement Classification
What does the Requirement Classification stand for?

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 witch 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.

**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).
Fit Gap Analysis and Delta Design – Step 3
Detailed Requirement Documentation

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

Step 3: How to do detailed Requirement documentation?

FB Role: Business Analyst
Functionality: Requirement Management, My Requirements
Procedure:
- After the Fit/Gap workshops, rework the Requirements in status ‘Draft’. When ready, set the Requirement status ‘To be Approved’
- In case the Requirements have been documented in xls, or another system, upload to On premise SAP Solution Manager. Assign the Solution Documentation. Check and rework the Requirements. When ready, set the Requirement status ‘To be Approved’

Advantage:
- Efficient communication between functional and technical experts: Detailed Requirement description and business context helps the Architect and Development to better understand the Requirements
Requirement

Practical Tips

- Define and discuss within project team how to use priority, category and other attributes across project teams!

- Decide on categories before starting requirements gathering in order to simplify searching and filtering (relevant for Mass Change, SRD); reuse them in WP and WI as well

- Add meaningful names and description to requirements – especially when working on bigger teams!

- If required also document Fit requirements for holistic documentation
  -> but define how this shall be handled otherwise the amount of requirements is growing too fast

- Define the Sign-off process together with the customer to avoid a bottleneck at the beginning of the configuration/development activities
Focused Build Analytics
Available dashboards to support related Release and Project phases

Release Phases
- Prepare
- Build
- Test
- Deploy

Project Phases
- Discover & Prepare
- Explore
- Realize
- Deploy

Dashboard usage
- Program Manager
- Project Manager
- Architect
- Test Manager
- Release Manager

Solution Readiness Dashboard
Test Suite Dashboard
Release Dashboard
Requirements Reporting in Solution Readiness Dashboard

Example: Unassigned Requirements

- Status overview with aggregated numbers
- Detailed list view with filter options
- Navigation to each single Requirement for all details
Fit Gap Analysis and Delta Design – Step 4
Define Initial Backlog

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
   - Prioritization according to agreed criteria (e.g. Business Value, Criticality)

Step 4: How to define the Initial Backlog?

FB Role: Release/Project Manager, Architect
Functionality: Mass Change Operation

Procedure:
- Select Requirements in Status 'To be Approved'
- Sort Requirements according Priority, Value or Effort Points
- Check and adapt the values
- Sign-off: Approve those Requirements to be part of the Release
- Postpone those Requirements to be part of the next Release

Reporting:
- Solution Readiness Dashboard (Unassigned Requirements, Requirements Management)

Advantages:
- Efficient prioritization via filter and sort functionality
- Efficient approval process via mass change functionality

Note:
- In fix price/waterfall projects, the Initial Backlog is a fixed list of Requirements defining the complete content of a Release. So you need only one Release for the whole project and you could plan the complete project already at this point in time.
- In agile projects, the Initial Backlog might change. So the Architects check the Initial Backlog each wave again. They plan their product backlog according the current prioritization. So typically you have a Release for each wave.
Define Initial Backlog

Example: Use Mass Change Operations to Approve Requirements

Select all Requirements to be part of the Release

Change the Requirement Status to ‘Approved’
Define Initial Backlog
Example: Use Mass Change Operations to Prioritize Requirements

### Mass Change for Requirement

<table>
<thead>
<tr>
<th>Solution:</th>
<th>Branch:</th>
<th>Scope Attribute:</th>
<th>Element:</th>
<th>ID:</th>
<th>Title:</th>
<th>Status:</th>
<th>Priority:</th>
</tr>
</thead>
<tbody>
<tr>
<td>AT F3 SP04 Solution</td>
<td>Design</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Approved</td>
<td></td>
</tr>
<tr>
<td>Value Points:</td>
<td>Effort Points:</td>
<td>Classification:</td>
<td>Category:</td>
<td>Work Package Assignment:</td>
<td>Owner:</td>
<td>Max Results:</td>
<td></td>
</tr>
</tbody>
</table>

### Sort Values

Sort Values

### Check and adjust Priority

Check and adjust Priority
Agile Project Execution with 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

**Step B**
Delta Design

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

5. **Delta Design**

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

   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

© 2019 SAP SE or an SAP affiliate company. All rights reserved. ǀ PUBLIC
Realize Phase - Plan Product Backlog
Release Planning: Roles and Responsibilities

1. Define Product Backlog

- Business Responsibility: Business Analyst, Project Manager
- Use Value points, effort and priority to rank
- Business Analyst, Project Manager

2. Prioritize Product Backlog

- Business Responsibility: Business Analyst, Project Manager
- Use Value points, effort and priority to rank
- Business Analyst, Project Manager

3. What would you like to have in the Release?

- Business Responsibility: Business Analyst, Project Manager
- Use Value points, effort and priority to rank
- Business Analyst, Project Manager

4. Analysis of Dependencies

- IT Responsibility: Architect

5. Estimate Product Backlog

- IT Responsibility: Architect

6. Scope Release: Build logical and technical packages

- IT Responsibility: Architect

Initial Backlog = High-level Release Plan based on Master Project

Product Backlog = Detailed Release Plan based on Build Project and Wave
Waves

Wave details

- A wave comprises a well-defined functional scope of work packages to define what needs to be done
- Starts with scope definition and a preparation time
  - Provides at least the functional specification required to start the first sprint of the wave
- Actual build execution is done in Sprints
- Execute functional integration testing
- Ends with the q-gate “Wave exit-criteria fulfilment” for passing the q-gate
- Optional: Release can be assigned to a Wave to allow Go-Live after the Wave

Legend
- ▲ = Q-Gate
- □ = Milestone

Work Package assigned to a Wave receiving milestones from Wave as default due dates
Agile Project Execution with Focused Build

### Highlevel Release Planning (Master Project)
- **Initial Backlog:** Requirements
  - Value: Effort
  - 10 - 6, 6 - 5
  - 23 - 3, 14 - 3
  - 9 - 3, 7 - 7
  - 34 - 3, 10 - 4
  - 39 - 2, 5 - 2
  - 45 - 2, 1 - 2
- **Maintain Requirements**
- **Pre-built system or pre-assembled solution**
- **Baseline Build**
- **Solution Validation**

### Detailed Release Planning (Build Project)
- **Product Backlog:**
  - **Work Packages**
  - **Value**: Effort
  - **Must**:
    - Wave 1
  - **Could**:
    - Wave 1
  - **Would**:
    - Wave 1
- **Architecture**
- **Sprints 1-3**
- **Demo/Test**
- **Detailed Release Planning (Build Project)**
- **Wave Planning**

### Wave 1
- **Wave 1**
  - **Architecture**
  - **Developers**
  - **Sprints 1-3**
  - **Demo/Test**
  - **Wave Planning**

### Wave 2
- **Wave 2**
  - **Architecture**
  - **Developers**
  - **Sprints 4-6**
  - **Demo/Test**
  - **Wave Planning**

### Support
- **Jump Start**
- **Prepare**
- **Explore**
- **Realize**
- **Plan**
- **Deploy**
- **Run**

© 2019 SAP SE or an SAP affiliate company. All rights reserved. | PUBLIC
Fit Gap Analysis and Delta Design – Step 5
Define Product Backlog

Step 5.1: How to define the Product Backlog?

FB Role: Project Manager, Architect
Functionality: Requirement Management
Procedure:
▪ Select Requirements in Status ‘Approved’ and in your responsibility, e.g. filter according process element or owner
▪ Create Work Packages and assign them to your Project
▪ Maintain Priority, Value or Effort Points

Procedure for Product Backlog Creation:
▪ In Mass Change Operation filter for Work Packages part of your project
▪ Assign Wave

Reporting:
▪ Solution Readiness Dashboard (‘To be Scoped’ Requirements, Work Packages), Mass Change Operation

Advantages:
▪ Good overview of Requirements in scope
▪ Efficient prioritization via filter and sort functionality

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

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
Create Work Package
Guidelines

Distribute Work Packages (WPs)

- Follow the organizational structure of the team, e.g. P2P, Logistics, Master Data, …
- Each team works on it’s Product Backlog which is the summary of the assigned Work Packages

Work Package tailoring

- You need to find a balance between detailed planning/description and efficiency/flexibility.
  - Firstly enable **functioning of standard business process** with high level process description as functional spec.
  - Then **work on gaps and fits** for this process during next wave. Here you do specific extensions to standard transactions or own developments
- A Work Package needs to be testable by an end- or key user (single functional and acceptance test)
- It needs to be realized in one wave
  -> Try to slice small Work Packages at the beginning. We’ve seen a lot of Work Packages couldn’t be finished in one wave (which is needed to be handed over to the next release)
Define Product Backlog
Example: Use Mass Change Operations to define the Scope of the first Wave
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
Step 5.2: How to do Delta Design?

FB Role: Architect
Functionality: My Work Packages
Procedure Delta Design:
- In Work Package upload Configuration Document or create Functional Specification

Reporting:
- Solution Readiness Dashboard (Functional Specifications)

Advantages:
- Automated Check, if KPI has been maintained by Project Manager. Solution Readiness Dashboard > tile Functional Specification > bar ‘Milestone Missing’
- Project Manager benefits from automated reporting on Functional Specifications, e.g. Overdue or Completed
Create Work Package

Guidelines

Maintain Work Packages

- Link to a process structure or library element
  - To automatically inherit the Process context, create Work Packages 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

- To ease filtering, establish a nomenclature for all Work Package Titles

- Assign at least one Functional Specification or Configuration Guide

- Subdivide into several Work Items in case the work is distributed between project members or dev. Teams
  - Don’t include too many developers in one Work Item, only in case they share the same timeline and effort
Check Documentation in Solution Readiness Dashboard
Example: Functional Specification

Each Work Package needs to have a Functional Specification or Configuration document. Non-technical Work Packages don’t need to have these documents.

Percentage of all your functional specifications according to status and due date

Drill down to display detailed information

Note: Dates for Milestones and Q-Gates are set by the central Project Management Organization in alignment with the customer. They are maintained in the Project Management functionality of SAP Solution Manager. They are the default values for the Work Packages/Items from there synchronized to the Solution Readiness Dashboard.
Functional Specification Tile

Percentage overview on Work Packages according to KPI definition for Functional Specification and related due dates.

**Completed:** Required document available and released

**To be done:** Required document does not meet KPI definition but Due Date has not been reached yet.

**Milestone Missing:** Date of related milestone or milestone is missing in project for assigned Wave

**Overdue:** Required document does not meet KPI definition and Due Date is in the past.

Click on bar allows drill down to list of relevant Work Packages
Fit Gap Analysis and Delta Design – Step 8
Verify & Accept

Step 6: How to do verification and sign-off of Delta Design?

FB Role: Architect, Customer
Functionality: My Work Packages
Procedure:
- Upload Functional Specification (drag and drop)
- Set the Document Status ‘In Review’
- Approver/Customer gets notified via e-mail
- Sets document in status ‘Released’

Reporting:
- Solution Readiness Dashboard (Functional Specifications, Work Packages, Current Wave Progress)

Advantages:
- Project Manager benefits from automated reporting on Functional Specifications
- Progress tracking of all Work Packages
Work Package Tile

Number of Work Packages assigned to selected Project
Aggregated view on Work Packages per Category and Wave with drill down to details
Current Wave Progress Tile

The actual completion rate in percentage across all work packages that are in scope of the current wave.

The Schedule tab displays the progress of the work packages of a selected wave. If you want to view a more detailed view of the current wave progress, you can also display the progress for work items per sprint by choosing Current Wave Progress for Work Items.

The KPIs tab shows the status of the documents for the work packages of the selected wave.
Realize Phase - Plan Sprint
Project Management: Release Sprint

To be able to start development and process a Work Item...

... the current Sprint needs to be released

Note: In case you release a Wave all underlying Sprints are automatically released. So if you want to work Agile, keep the Wave status ‘Created’ and release the Sprints only when the developers shall start to process their Work Items.
Sprints

Sprint details

- A sprint comprises a well-defined scope of work items to define how to do it
- Starts with sprint backlog definition and prioritize the work items
- Provides technical design documents and software developments for review in show and tell sessions / sprint reviews
- Unit test to confirm work item is completed
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>Non-functional</td>
<td>Non-functional</td>
</tr>
<tr>
<td>Non-functional</td>
<td>Non-functional</td>
<td>Non-functional</td>
<td>Non-functional</td>
<td></td>
</tr>
</tbody>
</table>
Create Work Items
Guidelines

A Work Item needs to be implementable in one sprint (~1-3 weeks)

You can have following relations between Work Package and Work Item:

- **1:1.** In case just one developer or consultant takes the action to fulfill the Work Package

- **1:n.** In case you want to:
  - Address different organizations and distribute required tasks (outsourced partners and internal organization)
    - Configuration goes to SAP consultant who is a member of the project
    - Frontend development goes to outsourced partner
    - Backend development goes to internal organization
  - Address different types of activities:
    - Configuration goes to consultant who is a member of the project
    - Frontend development goes to developer1 who is a member of the project
    - Backend development goes to developer2 who is a member of the project
Work Item Tile

Number of Work Items assigned to selected Project
Aggregated view on Work Items per Category and Classification with drill down to details
Technical Design Tile

Percentage overview on Work Items according to KPI definition for Technical Design and related due dates.

**Completed:** Required document available and released

**To be done:** Required document does not meet KPI definition but Due Date has not been reached yet.

**Milestone Missing:** Date of related milestone or milestone is missing in project for assigned Sprint

**Overdue:** Required document does not meet KPI definition and related Due Date is in the past.

Click on bar allows drill down to list of relevant Work Items
## Development Tile

Percentage of Work Items according to status and due date.

**Completed:** The Work Items have the status To Be Tested or later and the related Due Date has not been reached yet.

**To be done:** Work Items don't have the status To Be Tested yet and the related Due Date has not been reached yet.

**Milestone Missing:** Date of related milestone or milestone is missing in project for assigned Sprint

**Overdue:** The Work Items don't have the status To Be Tested or later and the related Due Date is in the past.

<table>
<thead>
<tr>
<th>Work...</th>
<th>Type</th>
<th>Title...</th>
<th>Status</th>
<th>WI Classific...</th>
<th>Cat...</th>
<th>Work Package</th>
<th>Wave</th>
<th>Dev...</th>
<th>Tester</th>
<th>Dev...</th>
<th>Pr...</th>
</tr>
</thead>
<tbody>
<tr>
<td>8000001</td>
<td>GC</td>
<td>WI.AssignStruct8_01_oek</td>
<td>Succ...</td>
<td>Workflow</td>
<td>Una...</td>
<td>WP.AssignStruct13</td>
<td>Wav...</td>
<td>Kata...</td>
<td>Kata...</td>
<td>4: Low</td>
<td></td>
</tr>
<tr>
<td>8000001</td>
<td>NC</td>
<td>Wi_test_02</td>
<td>Succ...</td>
<td>N/A</td>
<td>Una...</td>
<td>WP for Wi test</td>
<td>Wav...</td>
<td>OST...</td>
<td>OST...</td>
<td>2: High</td>
<td></td>
</tr>
<tr>
<td>8000001</td>
<td>GC</td>
<td>Wi_test_01</td>
<td>Succ...</td>
<td>N/A</td>
<td>Una...</td>
<td>WP for Wi test</td>
<td>Wav...</td>
<td>OST...</td>
<td>OST...</td>
<td>2: High</td>
<td></td>
</tr>
<tr>
<td>800001</td>
<td>GC</td>
<td>WI for testing SP04 21</td>
<td>To B...</td>
<td>N/A</td>
<td>Appli...</td>
<td>ST-OST 200 SP 04 - Proj...</td>
<td>Wav...</td>
<td>OST...</td>
<td>FB...</td>
<td>2: High</td>
<td></td>
</tr>
</tbody>
</table>
Unit Test Tile

Percentage of Work Items according to status and due date.

**Completed:** Work Items have the status ‘Successfully Tested’ or later

**To be done:** Status is not yet ‘Successfully Tested’ and the related Due Date has not been reached yet.

**Milestone Missing:** Date of related milestone or milestone is missing in project for assigned Sprint

**Overdue:** Status is not yet ‘Successfully Tested’ and the related Due Date is in the past.
Summary
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 months
- 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
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
Thank you.