SAP Solution Manager - Focused Build
Project Structure and Release Management options

Frank Jungmann, Reiner Markheiser, SAP SE
October 2021
Disclaimer

The information in this presentation is confidential and proprietary to SAP and may not be disclosed without the permission of SAP. Except for your obligation to protect confidential information, 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 presentation or any related document, or to develop or release any functionality mentioned therein.

This presentation, or any related document and SAP’s strategy and possible future developments, products and or 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 presentation is not a commitment, promise or legal obligation to deliver any material, code or functionality. This presentation 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. This presentation is for informational purposes and may not be incorporated into a contract. SAP assumes no responsibility for errors or omissions in this presentation, except if such damages were caused by SAP’s intentional 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.
Frank Jungmann (frank.jungmann@sap.com)
Speaker

Frank Jungmann works as Chief Support Architect for SAP Solution Manager and Focused Build. As product owner for Change and Release Management he is responsible for innovations, design and continuous improvement of applications and functions in this area.

Frank started at SAP in 1996. Before joining the SAP Support team in 2008, he worked as Consultant for SAP Germany in the area of CRM Service as well as Plant Maintenance and Customer Service.

Frank has a degree in Business Administration with focus on industry from the Berufsakademie Mannheim.
Reiner Markheiser (reiner.markheiser@sap.com)
Speaker

Reiner Markheiser works as Support Architect for Application Lifecycle Management. As product owner he is responsible for innovations in various application lifecycle management (ALM) products such as Focused Build and Cloud ALM.


Prior to SAP, he worked at SerCon GmbH / IBM Global Services as a Senior Consultant and KOMMOSOFT GmbH in Mannheim.

He has a degree in Industrial Engineering and Business Administration with focus on Information Technology from the University of Applied Sciences in Mannheim and Ludwigshafen.
Agenda

Objective

Focused Build overview

Default options regarding mapping between Project/Waves and Releases

Challenges and additional options
Objective
Project Structure and Release Management options in Focused Build

Objective

To be clarified for planned implementation project / program:

• Scope / Processes to be implemented
• Solution Landscape that is part of your implementation project / program
• Deployment / release strategy (e.g. incremental vs. big bang approach)
• Project methodology (e.g. agile vs. waterfall)
• Project structure (workstreams / development teams)
• Planned project timelines / Go-Live dates

Goal: Define Project structure in combination with Release Strategy considering the aspects above
Focused Build overview
Transparent Requirements-to-Deploy
Incremental deployment with constant feedback loops with the Business

**Releases** synchronize project Go-Lives and ensure continuous delivery and integration

**Project** to bundle deliverables
Multiple and parallel projects are possible

**Phases** ending with **Quality Gates**
Short prepare and scope, incremental realization

**Waves** ending with **Touch and Feel**
by the business (~ 8-12+ weeks)

**Sprints** with **Show and Tell** sessions to the business (~ 2-4 weeks)
Focused Build – Architecture Overview and Integration Model

Change Control Landscape

Process Management

- Process and Application Landscape
- Design Documents
- Branches
- Test Cases

Requirements / Work Packages / Work Items

- Requirements
- Work Packages
- Work Items

- Test Suite
- Test Plan
- Test Package

Release Management

Release Number and Cycle

Project Management

- Wave 1
- Sprint 1…n
- Task 1…n
- Milestone
- Q-Gate

Focused Build Dashboards

- Business Process Readiness Overview
- No. of work packages: 1367
- No. of work items: 4134
- Development: 12%
Available Project Types

**Single Project**

**Advantage**
- Individual Schedule of Projects, Phases, Wave and related Release Cycles per project possible
- Independent reporting via Solution Readiness Dashboard

**Disadvantage**
- Timeline and overall scheduling to be maintained for each project
- Only 1 user can edit project for planning and update activities at the same time
- Potential performance issues if project structure gets huge (e.g. > 1000 project elements)

**Main / Build Project**

**Advantage**
- Manage large projects in parallel with different build teams
- Aggregated status reporting on Main Project level
- Supports scaled agile principles
- Main project leads the timeline and overall scheduling
- Split work and responsibility to several sub project managers
- Ability to test across build projects (integration test)

**Disadvantage**
- All projects must be connected to same Change Control Landscape / Release Component

---

[SAP Recommendation]
Solution Readiness Dashboard – Main Project

Aggregated view for all Build-Projects assigned to Main Project

Aggregated data per Build Project

Select Build Project
Solution Readiness Dashboard – Build Project

Aggregated view for selected Build-Project

Aggregated data per Category
SAP Solution Manager 7.2
Introduction to Release Management

**Release & Deployment Management** is essential to manage successful deployment of all related changes into the productive environment.

**Release Management** defines dates and schedule for Releases.

**Release** content is determined by Projects, Requests for Changes, and related Change Documents.

The **Release Cycle** is a technical container and model for the release content.
Goal of Release Management
Synchronize Go-Live and further deployments to production

Reduce Risks, simplify Go-Live Process and decrease Test Efforts
Entities relevant for Release Management

- Logical Component Group
- Logical Component
- Branches
- Task List
  - "Minor Release"
- Task List
  - "Major Release"
- Minor Release Cycles
- Release/Change Cycles
Release Management Concept in Focused Build

Whenever something shall be deployed into the productive environment, it will be handled via a Release.

Work Packages are assigned to one of the waves within Project Management. Such a wave is associated with a certain release, either implicitly via Project Header or explicitly per Wave under Details → Basic Information. All Work Packages, assigned to the same release within 1 project, will be deployed together (Go-Live) at the end of the project or wave.
Default options regarding mapping between Project/Waves and Releases
## Define Project Management Structure and mapping to Releases

Example: Synchronous project, wave schedule and release deployment

### Standard Main / Build Project Structure:

<table>
<thead>
<tr>
<th>Main Project</th>
<th>Build Project 1</th>
<th>Build Project 2</th>
<th>Build Project 3</th>
<th>Release Cycle</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project Name</td>
<td>e.g. S/4HANA</td>
<td>e.g. Finance</td>
<td>e.g. Logistics</td>
<td>e.g. HCM</td>
</tr>
<tr>
<td>Solution Documentation Scopes</td>
<td>e.g. ‘Show All’</td>
<td>e.g. Finance</td>
<td>e.g. Logistics</td>
<td>e.g. HCM</td>
</tr>
<tr>
<td>Number of Weeks per Sprint</td>
<td>&lt;n&gt;</td>
<td>&lt;n&gt;</td>
<td>&lt;n&gt;</td>
<td>&lt;n&gt;</td>
</tr>
<tr>
<td>Number of Sprints per Wave</td>
<td>&lt;n&gt;</td>
<td>&lt;n&gt;</td>
<td>&lt;n&gt;</td>
<td>&lt;n&gt;</td>
</tr>
<tr>
<td>Number of Waves per Project</td>
<td>&lt;n&gt;</td>
<td>&lt;n&gt;</td>
<td>&lt;n&gt;</td>
<td>&lt;n&gt;</td>
</tr>
<tr>
<td>Start date of Project</td>
<td>&lt;date&gt;</td>
<td>&lt;date&gt;</td>
<td>&lt;date&gt;</td>
<td>&lt;date&gt;</td>
</tr>
<tr>
<td>Duration Discovery &amp; Prepare phase (days)</td>
<td>&lt;n&gt;</td>
<td>&lt;n&gt;</td>
<td>&lt;n&gt;</td>
<td>&lt;n&gt;</td>
</tr>
<tr>
<td>Duration Explore phase (days)</td>
<td>&lt;n&gt;</td>
<td>&lt;n&gt;</td>
<td>&lt;n&gt;</td>
<td>&lt;n&gt;</td>
</tr>
<tr>
<td>Duration Deploy phase (days)</td>
<td>&lt;n&gt;</td>
<td>&lt;n&gt;</td>
<td>&lt;n&gt;</td>
<td>&lt;n&gt;</td>
</tr>
<tr>
<td>Duration Run phase (days)</td>
<td>&lt;n&gt;</td>
<td>&lt;n&gt;</td>
<td>&lt;n&gt;</td>
<td>&lt;n&gt;</td>
</tr>
<tr>
<td>Project/Wave/Release Mapping</td>
<td>Wave &lt;n&gt;</td>
<td>Wave &lt;n&gt;</td>
<td>Wave &lt;n&gt;</td>
<td>Wave &lt;n&gt;</td>
</tr>
<tr>
<td></td>
<td>Wave &lt;n&gt;</td>
<td>Wave &lt;n&gt;</td>
<td>Wave &lt;n&gt;</td>
<td>Wave &lt;n&gt;</td>
</tr>
<tr>
<td></td>
<td>Wave &lt;n&gt;</td>
<td>Wave &lt;n&gt;</td>
<td>Wave &lt;n&gt;</td>
<td>Wave &lt;n&gt;</td>
</tr>
<tr>
<td></td>
<td>Wave &lt;n&gt;</td>
<td>Wave &lt;n&gt;</td>
<td>Wave &lt;n&gt;</td>
<td>Wave &lt;n&gt;</td>
</tr>
<tr>
<td></td>
<td>Wave &lt;n&gt;</td>
<td>Wave &lt;n&gt;</td>
<td>Wave &lt;n&gt;</td>
<td>Wave &lt;n&gt;</td>
</tr>
</tbody>
</table>
## Example for schedule of Releases, Projects, Waves and Sprints

**Use Case: Go-Live at the end of the Project**

### Phases of Release Cycles
- **Major Releases**

### Main Project
- **Phases**
  - **Waves**

### Build Project 1
- **Phases**
  - **Waves**
    - **Sprints**

### Build Project 2
- **Phases**
  - **Waves**
    - **Sprints**

---

### Disc. & Prepare

<table>
<thead>
<tr>
<th>JAN</th>
<th>FEB</th>
<th>MAR</th>
<th>APR</th>
<th>MAI</th>
<th>JUN</th>
<th>JUL</th>
<th>AUG</th>
<th>SEP</th>
<th>OCT</th>
<th>NOV</th>
<th>DEC</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td><strong>Prepare</strong></td>
<td><strong>Deploy</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### Explore

<table>
<thead>
<tr>
<th>Wave 1</th>
<th>Wave 2</th>
</tr>
</thead>
</table>

### Realize

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
</table>

### Deploy

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
</table>

### Example for schedule of Releases, Projects, Waves and Sprints

**Use Case: Go-Live at the end of the Project**

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
</table>

### Disc. & Prepare

<table>
<thead>
<tr>
<th>Wave 1</th>
<th>Wave 2</th>
</tr>
</thead>
</table>

### Explore

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
</table>

### Realize

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
</table>

### Deploy

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
</table>

### Define & Prepare

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
</table>

### Explore

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
</table>

### Realize

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
</table>

### Deploy

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
</table>

### Example for schedule of Releases, Projects, Waves and Sprints

**Use Case: Go-Live at the end of the Project**

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
</table>

### Disc. & Prepare

<table>
<thead>
<tr>
<th>Wave 1</th>
<th>Wave 2</th>
</tr>
</thead>
</table>

### Explore

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
</table>

### Realize

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
</table>

### Deploy

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
</table>

### Example for schedule of Releases, Projects, Waves and Sprints

**Use Case: Go-Live at the end of the Project**

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
</table>

### Disc. & Prepare

<table>
<thead>
<tr>
<th>Wave 1</th>
<th>Wave 2</th>
</tr>
</thead>
</table>

### Explore

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
</table>

### Realize

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
</table>

### Deploy

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
</table>

### Example for schedule of Releases, Projects, Waves and Sprints

**Use Case: Go-Live at the end of the Project**

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
</table>

### Disc. & Prepare

<table>
<thead>
<tr>
<th>Wave 1</th>
<th>Wave 2</th>
</tr>
</thead>
</table>

### Explore

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
</table>

### Realize

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
</table>

### Deploy

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
</table>

---

© 2021 SAP SE or an SAP affiliate company. All rights reserved. | PUBLIC
Example for schedule of Releases, Projects, Waves and Sprints
Use Case: Go Live per Wave

- **Release Schedule**
  - Major Release 1.0
  - Major Release 2.0

- **Main Project**
  - Phases
    - Waves

- **Build Project 1**
  - Phases
    - Waves
      - Sprints

- **Build Project 2**
  - Phases
    - Waves
      - Sprints

![Diagram showing the schedule of releases, projects, waves, and sprints with timelines for major releases, main project phases, build project 1 and build project 2, and quality gates.](diagram.png)
Challenges and additional options
Challenges

Uncertainties in Project execution

- Actual effort higher than planned effort
- Delay due to dependencies to other workstreams / teams
- Fluctuation in project team
- Scope changes, …

→ Duration of project to be extended, Go-Live to be postponed.

Business is expecting further ongoing innovations / enhancements after 1\textsuperscript{st} Go-Live

→ Release Management is required to

- ensure minimized impact (regressions) on productive processes
- align possible downtimes with Business for future Release Go-Lives across projects
Additional options

- **Use Case 1**
  - Synced project timelines and Go-Lives, changing the interval for Waves and Release Cycles

- **Use Case 2**
  - Different project timelines and go-lives per LoB/functional area running as part the same program for the same landscape

- **Use Case 3**
  - Fixed Release schedule while running small independent projects

- **Use Case 4**
  - Independent projects on the same landscape
Example for schedule of Releases, Projects, Waves and Sprints

Use Case: Go Live per Wave while changing the deployment interval

- **Release Schedule**
  - Major Release 1.0
  - Major Releases 2.0 to n.n

- **Main Project**
  - Phases
    - Waves

- **Build Project 1**
  - Phases
    - Waves
      - Sprints

- **Build Project 2**
  - Phases
    - Waves
      - Sprints

- **Quality Gates**

- **Go Live Dates**
  - Wave 1: 1.0
  - Wave 2: 2.0
  - Wave 3: 3.0
  - Wave 4: 4.0

- **Release Number**
  - n.n
Additional options

- **Use Case 1**
  - Synced project timelines and Go-Lives, changing the interval for Waves and Release Cycles

- **Use Case 2**
  - Different project timelines and go-lives per LoB/functional area running as part the same program for the same landscape

- **Use Case 3**
  - Fixed Release schedule while running small independent projects

- **Use Case 4**
  - Independent projects on the same landscape
Example for schedule of Releases, Projects, Waves
Common start with synced wave schedule and independent release deployment per build project

- **Release Schedule**
  - Major Releases

- **Main Project**
  - Phases
    - Waves

- **Build Project 1**
  - Phases
    - Waves

- **Build Project 2**
  - Phases
    - Waves

- **Build Project 3**
  - Phases
    - Waves

© 2021 SAP SE or an SAP affiliate company. All rights reserved. | PUBLIC
Use Case 1
– Synced project timelines and Go-Lives, changing the interval for Waves and Release Cycles

Use Case 2
– Different project timelines and go-lives per LoB/functional area running as part the same program for the same landscape

Use Case 3
– Fixed Release schedule while running small independent projects

Use Case 4
– Independent projects on the same landscape
Example for schedule of Releases, Projects, Waves
Small projects with simple wave schedule to allow handover of WPs (leftovers) to next release

- **Release Schedule**
  - Major Releases

- Finance
  - Phases
    - Waves

- Purchasing
  - Phases
    - Waves

- Logistics
  - Phases
    - Waves
Additional options

- **Use Case 1**
  - Synced project timelines and Go-Lives, changing the interval for Waves and Release Cycles

- **Use Case 2**
  - Different project timelines and go-lives per LoB/functional area running as part the same program for the same landscape

- **Use Case 3**
  - Fixed Release schedule while running small independent projects

- **Use Case 4**
  - Independent projects on the same landscape
Example for schedule of Releases, Projects, Waves

Individual projects with 1 wave to allow handover of WPs (leftovers) to next release

- **Release Schedule**
  - Major Releases

- **Finance**
  - Phases
    - Waves

- **Purchasing**
  - Phases
    - Waves

- **Logistics**
  - Phases
    - Waves

Add new Wave with follow-up Release Cycle for projects expected to be not finished at Go-Live and do not set WP in status ‘Handover to Release’ for these projects.

→ Reassign WPs to next Wave / Release

© 2021 SAP SE or an SAP affiliate company. All rights reserved. | PUBLIC
Additional options II

- Options for reassignment in Release Management
  - Automatic reassignment
  - Manual reassignment in dialog / mass change
Automatically reassign when Release switch to "TEST"

- Automatically reassign open change documents to the successor release already, when the release is switched to phase “Test”.
- More agile processing
- Keep developing your increments also during release testing
Automatic Reassignment options

Release 1.0:
- Created
- Prepare
- Build
- Test
- Deployment Preparation
- Deploy
- Hypercare
- Operate
- Completed

Optional automatic Reassignment

Mandatory automatic Reassignment

Release 2.0:
- Created
- Prepare
- Build
- Test
- Deployment Preparation
- Deploy
- Hypercare

Optional automatic Reassignment

Mandatory automatic Reassignment

Release 3.0:
- Created
- Prepare
- Build
- Test
- Deployment Preparation

Optional automatic Reassignment

Mandatory automatic Reassignment
Automatically reassign when Release switch to "TEST“

Phase dependency for reassignment for Focused Build Release

Optional early automatic reassignment when release phase switch to “Test”

Manual reassignment allowed

Reassignment not allowed

Mandatory automatic reassignment when release phase switch to “Hypercare”
Automatically reassign when Release switch to "TEST"

Status dependency for reassignment to a later release for Work Packages

Reassignment allowed

Reassignment not allowed

Handover to Release is the point of no return!
Automatically reassign when Release switch to "TEST"

Status dependency for reassignment to a later release for Work Items

Reassignment allowed

Created  In Development  To be tested  Successfully tested

Reassignment not allowed

Handed over to Release  Productive  Completed

Handover to Release is the point of no return!
Automatically reassign when Release switch to "TEST"

Status dependency for reassignment to a later release for Preliminary Work Items

As soon as Preliminary Import was requested the Work Item has to be finished within the current Release
Manual reassignment in dialogue

Details in Work package:
- Open the drop down for “Wave”
- Choose the new Wave you want to re-assign
- Save
Manual reassignment via mass change

Mass Change for Work packages:
- Select the work Packages you want to re-assign
- Press the button
- Check Wave and choose the new Wave you want to re-assign
- Press button
Automatically reassign when Release switch to "TEST"

Early Reassignment
When Early Reassignment is switched on, you can automatically reassign open change documents to the successor release already, when the release is switched to phase Test.

When Early Reassignment is switched off, the automatic reassignment of open change documents to the successor release is executed only, when the release is switched to phase Hypercare.

Note: Only change documents that have not reached status Handed over to Release can be reassigned to a successor release.
Automatically reassign when Release switch to "TEST"

Cross Project Reassignment
When Cross Project Reassignment is switched on, open change documents can be reassigned to the successor wave/release of a project different than the current one. An additional popup will appear to choose the target project and wave.

When Cross Project Reassignment is switched off, open change documents will be reassigned to the successor wave/release of the current project. No additional popup will appear.

Note: Only change documents that have not reached status Handed over to Release can be reassigned to a successor release.
Thank you.