Focused Build for SAP Solution Manager 7.2 (SP6)
Deployment & Release Management

Frank Jungmann, SAP SE
June 2020
Agenda

Introduction

Focused Build – TMS integration

Release Management concept in Focused Build

Release Planning

Track Release

Transport Concept
Transparent Requirements-to-Deploy
Release Management to synchronize projects and keep them under control

- Build & Functional Test
- Freeze
- Go-Live

Project 1

- Build & Functional Test
- Freeze
- Go-Live

Project 2

- Build & Functional Test
- Freeze
- Go-Live

Project 3

Joint Integration Test Phase

Scope
- Build
- Test
- Deploy
- Hypercare
- Operate

Release 1

Reduce Risks, Simplify Go-Live Process, and Decrease Test Efforts
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 build.
- **Waves** ending with **Touch and Feel** by the business (~ 8-12+ weeks)
- **Sprints** with **Show and Tell** sessions to the business (~ 2-4 weeks)
**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 build

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

- **Sprints** with **Show and Tell** sessions to the business (~ 2 weeks)

---

**Program Level**
- Release automation
- Automated reporting
- Pragmatic documentation handling
- Automated completion tracking

**Team Level**
- Test planning & execution automation
- Jira Integration, unit test automation, transport automation

---

© 2020 SAP SE or an SAP affiliate company. All rights reserved. | PUBLIC
Example of Project Structure
Build Project for Waterfall Approach

Project structure sample with 1 wave and 1 sprint

- Wave 1
  - Functional Specification Released
  - Technical Design Released
  - (Work Item) Build Started
- Sprint 1
  - Unit Test Completed
  - (Work Item) Build Completed
- Wave Milestones
  - Single Func. Test Completed
  - Single Functional Test
- Phases of Release Cycle
  - Prepare
  - Realize
  - Deploy
  - Run

- Phases of Project Phases
  - Prepare
  - Explore
  - Realize
  - Deploy
  - Run

- Wave
  - Functional Integration Test (Final)
  - Acceptance Test
  - Regression Test
  - Handover to Release

- Sprint
  - (Work Package) Build Finished
  - Unit Test

(Note. Point of NO RETURN for the Transports in these WP)
Example of Project Structure
Build Project for Agile development of a single release

Project structure sample with 1 waves, of 3 sprints

Phases of Release Cycle
Prepare → Explore → Realize → Build → Test → Deploy → Run

Wave 1
- W1 Scope defined
- Functional Spec released/available
- Single Functional Test (SFT) completed
- Work Package (WP) Build completed
- Acceptance Test

Sprints
- Sprint 1
  - Tech. Design Released
  - Build Started
  - Build Completed
  - Unit Test Completed
- Sprint 2
  - Unit Test
- Sprint 3
  - Unit Test

(Note. Point of NO RETURN for the Transports in these WP)

Handover to Release

Wave Milestones
- Milestone 1
- Milestone 2
- Milestone 3

Sprint Milestones
- Milestone 1
- Milestone 2
- Milestone 3
Example of Project Structure
Build Project for Agile development of a single release

Project structure sample with 2 waves, of 2 sprints

Phases of Release Cycle
Prepare | Explore | Realize | Deploy | Run

Wave 1
FS Released

Sprints
- Sprint 1
  - Tech. Design Released
  - Build Started
  - Build Completed
  - Unit Test Completed
- Sprint 2
  - SFT
  - WP Build Finished
- AT1
- FIT1 (Optional)

Wave 2

Sprints
- Sprint 1
  - SFT
  - SFT completed
- Sprint 2
  - SFT
  - WP Build Finished
- AT1
- FIT1 (Optional)

Functional Integration Test (Final)
Regression Test

(Not. Point of NO RETURN for the Transports in these WP)

Handover to Release

Sprint Milestones
- Unit Test

Wave Milestones
- SFT
- FIT1 (Optional)

Single Functional Test

© 2020 SAP SE or an SAP affiliate company. All rights reserved. I PUBLIC
# Recommended System Landscape for Focused Build

<table>
<thead>
<tr>
<th>System Landscape (TMS)</th>
<th>Logical Component Group</th>
<th>Branches</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Production</strong></td>
<td><strong>Log Comp Group ERP</strong></td>
<td><strong>Production</strong></td>
</tr>
<tr>
<td>PRD-100</td>
<td>Logical Component Production</td>
<td></td>
</tr>
<tr>
<td><strong>Development</strong></td>
<td>Logical Component Development</td>
<td><strong>Development</strong></td>
</tr>
<tr>
<td>DEV-100 QAD-100 PRE-100</td>
<td>Logical Component Design</td>
<td>Design</td>
</tr>
<tr>
<td><strong>Sandbox</strong></td>
<td>Logical Component Import</td>
<td>Import</td>
</tr>
<tr>
<td>SBX-100</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Focused Build – TMS integration
Focused Build methodology – Systems
Standard Workflow

SBX
- Approved
- In Realization
- Created
- Scoping
- Scope finalized
- To be Developed
- In Development
- Create Work Item

QAS
- Single Functional Test & AT
- To be tested
- Successfully tested
- Handover to release
- IT, RT

PRE
- Realization Completed
- Handed over to release
- Handover to release

PRD
- Productive
- Productive/ Completed

DEV
- Created
- In Development
- Unit Test
- Automated Transport
- Create Transports
- To be tested
- Successfully tested
- Handover to release
- Productive/ Completed
Focused Build - Transport Handling with ToC and Unit Test in QAS
Detailed Transport/Status Dependencies

* Transport of Copies

© 2020 SAP SE or an SAP affiliate company. All rights reserved. I PUBLIC
Work Item status flow with integrated TMS actions

Legend:
- **Transport of Copies**
- **Regular Transports**

### Work Item

- **Created**
- **In Development**
- **Release Tasks**
- **To Be Tested**
- **Successfully Tested**
- **Handed Over to Release** (Status set by WP)
- **Productive** (Status set by Import)

**Legend:**
- **Regular Transports**
- **Transport of Copies**

**Status Flow:**
- **DEV**
  - Created
  - In Development
  - Release Tasks
  - To Be Tested
- **QAS**
  - Successfully Tested
- **PRE**
  - Handled Over to Release (Status set by WP)
- **PRD**
  - Productive (Status set by Import)

**Steps:**
- Import in QAS
- Import in PRE
- Import in PRD
What happens with Work Package, Work Item and Requirement at Go-Live
Combined Productive and Completed status

- 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
Focused Build methodology
Standard Workflow at Go-Live

‘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.

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.
Release Management concept in Focused Build
Example for schedule of Releases, Projects, Waves and Sprints
Use Case: One Release per project, two Waves, and three Sprints per Wave

Phases of Release Cycles
- Major Releases

MASTER Project
- Phases
  - Waves

Build Project 1
- Phases
  - Waves
    - Sprints

Build Project 2
- Phases
  - Waves
    - Sprints

Quality Gates
Example for schedule of Releases, Projects, Waves and Sprints

Use Case: Two Releases per project, two Waves, and three Sprints per Wave

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

- **Master Project**
  - Phases
    - Waves

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

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

Year

- **Prepare**
- **Build**
- **Test**
- **Hypercare**
- **Operation**

- **Release Number**
- **Go-Live Date**

© 2020 SAP SE or an SAP affiliate company. All rights reserved. | PUBLIC
Release Management Concept in Focused Build

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

Usually Work Packages are assigned to one of the waves within Project Management. Such a wave is associated with a certain release. 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.
Release Phases - Introduction

The release has many different phases which all have a defined semantic meaning.

The phases of the release also control which activities are possible in the Work Packages and Work Items – related to the release,

- e.g. assignment to a release, import of transports, …
In the **Prepare** phase, the development of the release content takes place

- Work Packages will be processed and broken down into Work Items
- Changes will be implemented (with Workbench and Customizing Transports)
- In addition, at the end of a sprint the Single Functional Test of Work Packages will be performed within this phase and at the end of a wave the optional (early) Functional Integration and the UA Test will take place as well in the cycle phase. After the test plans are completed according the exit criteria’s defined at the beginning, a board consisting of Release Manager(s) and Test Manager(s) will decide on which Work Packages are supposed to go with the next release deployment. The supporting tool(s) for this decision are the Test Suite Dashboard – Traceability Matrix and the Mass Change Operations app.
Prepare Phase (2)

Test Plans in combination with Traceability Matrix to verify that everything is tested which shall be included into the release.

Mass Change Operations app for bulk changing status of Work Packages identified for the next release deployment.
In the build phase, the content of the release will be finally defined.

This is the time, when the release manager needs to make a decision about which Work Packages shall be part of the release deployment and which shall be taken out (e.g. postponed to next release). For the selection of Work Packages which could be part of the release Mass Change Operation app will be utilized:

Please refer to the slides of the previous phase regarding the completed Single Functional Test phase for the final scope of the release as well.
In the build phase, you are now switching the status of all Work Packages which are previously nominated for the next release deployment with the help of the Mass Change Operations app from Tested to Handed over to Release. **Please be aware of the fact this will be the Point of NO Return!** After setting the status ‘Handed over to Release’ there is just the way forward into the pre-production and finally into the productive system. **NO WAY BACK ANYMORE!**
– In the test phase, the testing of the release takes place; the content is finalized
– This is the final release test, which means the entire package needs to be validated and tested for functional and technical correctness before the import into the Production environment
– Usually this involves a whole series of tests including
  ▫ Regression Test
  ▫ Integration Test
– In order to have a valid test result, the Pre-Production system needs to be reset before the test phase
– In case of bugs or issues, a Defect Correction can be created to fix the errors
In the Deployment preparation phase, the technical steps to perform the cutover into the production system need to be done.

- This includes all required preparation steps – technical as well as non-technical.
- Typically the Deployment preparation is a short phase which is directly followed by the Deployment phase (e.g. as part of a weekend).
Deployment Phase

- The Deployment phase includes the “big day” – the actual technical cutover of the entire release into the production environment.
- This means all transports will be imported into the production environment.
- The activity to trigger the import will be done by an Administrator or Technical Release Manager from within SAP Solution Manager.
- The content of the package will be calculated automatically by SAP Solution Manager and imported in the correct sequence.
After the Deployment there is a special phase called “Hypercare”

This is due to the fact that usually shortly after a Go-Live, the number of incidents increases significantly.

Sometimes this is due to missing information about the new functionality, wrong documentation or bugs which have not been detected and solved before.

During this time, there is still a high attention on the release and a consistent monitoring of the incident queue.

Usually this phases ends after a few days or weeks – then the release will be formally handed over from the Project Team to the IT Operation Team, which ensure production support.
Once Hypercare phase is completed you can switch the cycle phase to Operation which means that this release is now productively operated.
Release Planning
Plan Release

Focused Build - Release Manager

Release Management
Mass Change Operations
My Defect Corrections
Administration Cockpit
Change Control Management
Release Dashboard Reporting
Release Planning

SAP Solution Manager IT Service Management

Release Planning

Create

1. FB_Release_Oto
2. Major Release 1.0
3. Major Release 2.0
4. Major Release 3.0

Landscape / Release Version
Status
Go-Live
Branch
Cycle Description

Quarter 3
Quarter 4
Quarter 1

January

Major
Track Release
Focused Build – Release Dashboard
Main features (1)

Release Dashboard

Features
- Drill down from top level dimensions down to the transport request level
- Breadcrumb navigation to show current context
- New filter options for Work Packages, Defect Corrections and transport requests
- Get status ratings to evaluate the completeness of releases
- Find Work Items and transport requests, which are not compliant to the Focused Build release building process

Benefits
Effective release tracking
Easily identify release issues
Perform detailed analysis
Focused Build – Release Dashboard
Main features (2)

**Release Dashboard**

**Features**

1. Drill down from top level dimensions down to the transport request level
2. Filter options for Work Packages, Defect Corrections and transport requests
3. Status ratings
4. Evaluate Work Items and transport requests, which are not compliant to the Focused Build release building process.

**Benefits**

Effective release tracking
Easily identify release issues
Perform detailed analysis
Transport Concept
Batch Import

**SE38**

Program name: /SALM/BATCH_IMPORT_TRIGGER

In the scope of Focused Build it is used for import in all kinds of systems. The program can be scheduled on a regular basis via the section 'Scheduling Option'.
Batch Import

Select the Release or Cycle you want to perform the Import for

<table>
<thead>
<tr>
<th>Date</th>
<th>Cycle Type</th>
<th>Number</th>
<th>Release No</th>
<th>Transaction Description</th>
<th>Solution Landscape Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>20.11.2016</td>
<td>Minor Release</td>
<td>80000093104</td>
<td>1.1.0</td>
<td>CORPORATE_SOLUTION_FocusedBuild Release</td>
<td>CORPORATE_SOLUTION_FB1</td>
</tr>
<tr>
<td>26.11.2016</td>
<td>Major Release</td>
<td>80000005559</td>
<td>1.0.0</td>
<td>RELEASETEST_SUB_1 Release 1.0</td>
<td>RELEASETEST_SUB_1</td>
</tr>
<tr>
<td>03.12.2016</td>
<td>Minor Release</td>
<td>80000005585</td>
<td>1.1.0</td>
<td>BATCHING_SUB_1 Release 1.1</td>
<td>BATCHING_SUB_1</td>
</tr>
<tr>
<td>04.12.2016</td>
<td>Minor Release</td>
<td>80000005156</td>
<td>2.1.0</td>
<td>CORPORATE_SOLUTION_SUB_1 Release 2.1</td>
<td>CORPORATE_SOLUTION_SUB_1</td>
</tr>
<tr>
<td>10.12.2016</td>
<td>Minor Release</td>
<td>80000005293</td>
<td>1.1.0</td>
<td>RELEASETEST_SUB_1 Release 1.1</td>
<td>RELEASETEST_SUB_1</td>
</tr>
<tr>
<td>25.12.2016</td>
<td>Minor Release</td>
<td>80000005462</td>
<td>2.2.0</td>
<td>CORPORATE_SOLUTION_SUB_1 Release 2.2</td>
<td>CORPORATE_SOLUTION_SUB_1</td>
</tr>
<tr>
<td>31.12.2016</td>
<td>Minor Release</td>
<td>80000005560</td>
<td>1.1.0</td>
<td>RELEASETEST_SUB_1 Release 1.1</td>
<td>RELEASETEST_SUB_1</td>
</tr>
<tr>
<td>01.01.2017</td>
<td>Major Release</td>
<td>80000001405</td>
<td>1.0.0</td>
<td>RELEASETEST_SUB_DS Release 1.0</td>
<td>RELEASETEST_SUB_2</td>
</tr>
<tr>
<td>08.01.2017</td>
<td>Major Release</td>
<td>80000001593</td>
<td>2.0.0</td>
<td>RELEASETEST_SUB_DS Release 2.0</td>
<td>RELEASETEST_SUB_2</td>
</tr>
<tr>
<td>14.01.2017</td>
<td>Minor Release</td>
<td>80000009294</td>
<td>1.2.0</td>
<td>RELEASETEST_SUB_1 Release 1.2</td>
<td>RELEASETEST_SUB_1</td>
</tr>
<tr>
<td>15.01.2017</td>
<td>Major Release</td>
<td>80000009175</td>
<td>1.0.0</td>
<td>CORPORATE_SOLUTION_SUB_1 Release 1.0</td>
<td>CORPORATE_SOLUTION_SUB_1</td>
</tr>
<tr>
<td>06.02.2017</td>
<td>Major Release</td>
<td>80000009207</td>
<td>2.0.0</td>
<td>RELEASETEST_SUB_1 Release 2.0</td>
<td>RELEASETEST_SUB_1</td>
</tr>
<tr>
<td>04.02.2017</td>
<td>Minor Release</td>
<td>80000009361</td>
<td>1.2.0</td>
<td>RELEASETEST_SUB_1 Release 1.2</td>
<td>RELEASETEST_SUB_1</td>
</tr>
</tbody>
</table>
Batch Import

Possibility to use Multi-select or even Wild Card *

AGS_WORK_CUSTOM settings necessary
### Batch Import

Possibility to use the task list as selection criteria

<table>
<thead>
<tr>
<th>Cycle Type</th>
<th>Go-Live</th>
<th>Task ID</th>
<th>Number</th>
<th>Release No</th>
<th>Transaction Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Minor Release</td>
<td>25.11.2018</td>
<td>8000000032</td>
<td>0000036689</td>
<td>1.1.0</td>
<td>ZER0_SINE_MKHL Release 1.1</td>
</tr>
<tr>
<td>Minor Release</td>
<td>25.11.2018</td>
<td>8000000261</td>
<td>0000002973</td>
<td>1.2.0</td>
<td>CUTOVER YAS 1.2 DEV</td>
</tr>
<tr>
<td>Minor Release</td>
<td>10.09.2018</td>
<td>8000000302</td>
<td>0000010189</td>
<td>1.1.0</td>
<td>Agile_Sub 2 Release 1.1</td>
</tr>
<tr>
<td>Minor Release</td>
<td>13.04.2020</td>
<td>8000000272</td>
<td>0000059274</td>
<td>1.2.0</td>
<td>ACCEPTANCE_TEST_SUB_44 Release 1.2</td>
</tr>
<tr>
<td>Minor Release</td>
<td>17.01.2021</td>
<td>8000000053</td>
<td>0000006051</td>
<td>1.1.0</td>
<td>ACCEPTANCE_TEST_SUB_1 Release 4.1</td>
</tr>
<tr>
<td>Phase Cycle</td>
<td>00.09.0008</td>
<td>8000000261</td>
<td>0000000461</td>
<td></td>
<td>Maintenance with Retrofit</td>
</tr>
<tr>
<td>Phase Cycle</td>
<td>00.09.0008</td>
<td>8000000272</td>
<td>0000000462</td>
<td></td>
<td>Implementation (Retrofit Target)</td>
</tr>
<tr>
<td>Phase Cycle</td>
<td>00.09.0008</td>
<td>8000000045</td>
<td>0000010189</td>
<td></td>
<td>SW Implementation</td>
</tr>
<tr>
<td>Phase Cycle</td>
<td>00.09.0008</td>
<td>8000000044</td>
<td>000001391</td>
<td></td>
<td>SW Maintenance</td>
</tr>
<tr>
<td>Phase Cycle</td>
<td>00.09.0008</td>
<td>8000000046</td>
<td>000001515</td>
<td></td>
<td>CERAR Test Cycle 1</td>
</tr>
<tr>
<td>Phase Cycle</td>
<td>00.09.0008</td>
<td>8000000047</td>
<td>000001515</td>
<td></td>
<td>CERAR Test Cycle 2</td>
</tr>
<tr>
<td>Phase Cycle</td>
<td>00.09.0008</td>
<td>8000000073</td>
<td>000003262</td>
<td></td>
<td>CERAR Maintenance Cycle</td>
</tr>
<tr>
<td>Phase Cycle</td>
<td>00.09.0008</td>
<td>8000000076</td>
<td>000003263</td>
<td></td>
<td>CERAR Project Cycle 1 - Deploy Production</td>
</tr>
<tr>
<td>Phase Cycle</td>
<td>00.09.0008</td>
<td>8000000077</td>
<td>000003263</td>
<td></td>
<td>CERAR Project Cycle 2 - Deploy Development</td>
</tr>
<tr>
<td>Phase Cycle</td>
<td>00.09.0008</td>
<td>8000000086</td>
<td>000003594</td>
<td></td>
<td>CERAR Check - Maintenance Cycle</td>
</tr>
<tr>
<td>Phase Cycle</td>
<td>00.09.0008</td>
<td>8000000086</td>
<td>000003594</td>
<td></td>
<td>CERAR Check - Maintenance Cycle</td>
</tr>
<tr>
<td>Phase Cycle</td>
<td>00.09.0008</td>
<td>8000000086</td>
<td>000003594</td>
<td></td>
<td>CERAR Check - Maintenance Cycle</td>
</tr>
<tr>
<td>Phase Cycle</td>
<td>00.09.0008</td>
<td>8000000086</td>
<td>000003594</td>
<td></td>
<td>CERAR Check - Maintenance Cycle</td>
</tr>
<tr>
<td>Phase Cycle</td>
<td>00.09.0008</td>
<td>8000000086</td>
<td>000003594</td>
<td></td>
<td>CERAR Check - Maintenance Cycle</td>
</tr>
<tr>
<td>Phase Cycle</td>
<td>00.09.0008</td>
<td>8000000086</td>
<td>000003594</td>
<td></td>
<td>CERAR Check - Maintenance Cycle</td>
</tr>
<tr>
<td>Phase Cycle</td>
<td>00.09.0008</td>
<td>8000000086</td>
<td>000003594</td>
<td></td>
<td>CERAR Check - Maintenance Cycle</td>
</tr>
<tr>
<td>Phase Cycle</td>
<td>00.09.0008</td>
<td>8000000086</td>
<td>000003594</td>
<td></td>
<td>CERAR Check - Maintenance Cycle</td>
</tr>
<tr>
<td>Phase Cycle</td>
<td>00.09.0008</td>
<td>8000000086</td>
<td>000003594</td>
<td></td>
<td>CERAR Check - Maintenance Cycle</td>
</tr>
<tr>
<td>Phase Cycle</td>
<td>00.09.0008</td>
<td>8000000086</td>
<td>000003594</td>
<td></td>
<td>CERAR Check - Maintenance Cycle</td>
</tr>
<tr>
<td>Phase Cycle</td>
<td>00.09.0008</td>
<td>8000000086</td>
<td>000003594</td>
<td></td>
<td>CERAR Check - Maintenance Cycle</td>
</tr>
<tr>
<td>Phase Cycle</td>
<td>00.09.0008</td>
<td>8000000086</td>
<td>000003594</td>
<td></td>
<td>CERAR Check - Maintenance Cycle</td>
</tr>
</tbody>
</table>

© 2020 SAP SE or an SAP affiliate company. All rights reserved. | PUBLIC
Batch Import

Needs to be flagged for Imports into PRD systems

- Include “orphan” transports that were created via Task list
- Only give report results – no commit work
- Only import Transport of Copies
Batch Import

- Enable automatic job re-scheduling
- Will wait XX min after the last job was finished before next job is scheduled to avoid “overtakes”
- Put your own naming for easier finds when searching for jobs

- Activate DGP check for import (will be switched on automatically when “Import into Production Systems” is flagged
- Access to additional checks beyond down grade protection such as Cross-Reference check, ABAP test cockpit or custom checks
- Skip transports which are stopped by DGP check and go on (no best practice)
Batch Import

- Enable automatic E-Mail notification
  - Get informed about possible issues
  - Get return code information
  - Maintain the relevant address
Batch Import

- Check for change document relations
  - Is predecessor of
  - Is parallel to
  - Is successor of
- Only import if all relevant transports of dependent change documents including Defect Corrections inside the document flow of a Work Package are valid for import
- Check document relation also available on WP level (not possible simultaneous with WI check)
Thank you.