Specification Acceptance Checklist

Contents

Purpose

The purpose of this checklist is to fast-track the creation and review of Specifications by describing the minimum set of acceptance criteria required for such a Specification to meet before it is accepted by the Technical team for implementation. By describing up-front the standards and expectations placed by the Technical team upon the authors of Specifications, we hope to speed up delivery of specs and their reviews, and to reduce rework with the goal to meet project timeline and quality objectives.

Assumption

The overriding requirement is for the Specification to be completed with enough detail so that a technically sound solution can be implemented.

Fitness for Purpose

The Specification serves several purposes that are equally important for the document to be acceptable. The specification must describe:

- **Why** the solution was designed in the way described in the document; and
- **How** it will address the gap between a genuine Syensqo requirement and SAP standard; and
- **What** is required to be developed to meet the requirements of the business process.

Specifications communicate with two different and equally important audiences:

- Project team members, such as developers who must build the solution you have in mind, Testers who need to validate it, etc.
- Support and Operations people who must support this in future, resolve problems, extend it after go-live, and otherwise maintain it for 10 years or more.

Both of these groups are important customers, and their needs must be taken into account when writing Specifications.

With the above in mind, these items must be adhered to :

- A Specification should not focus on the implementation details of how you feel the development should be completed, i.e. it should not contain pseudo-code.
- The author has not injected actual code into the spec. This creates redundancy that becomes out of date with the actual code in the system, and developers are instructed to ignore this.
- Avoid huge amounts of implementation detail in the documentation, which someone must read and keep updated over time. Instead, talk to the developer.
- Describes exactly one development, and does not contain multiple developments or developments of different types (e.g. describes both an interface and the workflow it triggers)
- Content aligns with all applicable ERP Rebuild standards, and explains the reasons for any deviations where they occur.
- Check that the Specification describes how the functional gap identified in SAP Standard will be addressed. For example, rather than saying ‘Users will be able to change data’, specify how they will be able to change data, e.g. ‘A UI app” will be created for users to change data’.
- The specification should point to what type of artifacts will need to be created, e.g. BAdI implementations should specify the BAdI provided by SAP, etc.

- Any dependency on custom data dictionary objects or new custom data dictionary objects have been identified and created. If not, the corresponding functional specification for these objects is indicated.
- Authorization information (groups/objects/parameters...) are specified.
- The author has not left any sections blank or with the placeholder text still showing. Where a section is not applicable, it must be explicitly marked as such.
- Assumptions are documented clearly, and must be validated before development work starts to avoid ambiguity and rework.
- Test data and test steps are to be verified by the Author before documenting in Specifications, this is to ensure that the system and configuration prerequisites are ready for development work.
- Besides positive testing, negative testing will also be required to ensure Error handlings are tested properly to improve program resiliency.

General Checks

- One or more Custom Development IDs have been allocated, except for Conversion Specifications which use the number assigned in the master data register.
- Development IDs are provided in the Specification.
- A reference to the relevant process or process step within Signavio has been inserted as a hyperlink. This is important so we have traceability on which business processes will be impacted if such objects for maintainability and supportability purposes.
- No spelling mistakes are present in the document, and any abbreviations are documented in the Glossary.
- Formatting is neat and tidy, and presents the specification in a professional light. An inconsistent mixture of font faces, colors, sizes, etc. is unprofessional and should be avoided.
- Embedded diagrams are encouraged, but the original source files used to construct the diagrams must also be included. This allows future authors to amend the diagram via the original source file as necessary. Ideally, the embedded Draw.io functionality in Confluence is used.
- Dependencies on other technical objects or business processes are identified, and dependent documents and specifications are available via hyperlinks.
- Key contacts are identified (functional owner, business owner, technical, etc.).
- Security is important and needs to be detailed by providing standard or required custom authorization objects.
- ALL config required for the process and development to work needs to be completed before development can start.
- Label for Pod and Release to be assigned to the spec in Confluence.
- Test cases need to be maintained in detail, like positive, negative tests and expected result.

Development Type Specific Checks

Different types of custom developments introduce additional requirements to consider when writing the Specification.

Application Job (custom)

- The Selection screen details need to be provided.
- Customizing requirements specified (views to be updated, etc).
- Exception process/handling must be defined.

Conversion

- Logical Source and Target Systems are identified.
- Processing Type is specified (i.e. Direct Input, BAPI, IDoc, Web Service, OData API, etc.)
- Standard or custom load program name/BAPI/IDoc is specified to support this functionality.
- Data Mapping is provided for mappings which are not obvious to a suitably-skilled and experienced developer
- All translation requirements are clearly defined (including data validation rules, data derivation/calculation and default values, if applicable)Input and/or Output file layouts are provided for all record types possible
- Transaction volume is specifiedExecution frequency is specifiedRestart/Recovery requirements have been defined
- Error handling requirements are specified, including alerting requirement and expected action on failure.
- Application log requirement if applicable to be specified.

Enhancement

- The main Fiori app/transaction/program being enhanced.
- The proposed enhancement mechanism has been provided. Please note that this can only be finalized with input from the Technical team.
- Customizing requirements specified (views to be updated, etc).
- Exception process/handling must be defined.
- Functional process should be explained to justify enhancement.
- Localisation / Regulatory requirements to be specified if applicable.

Form

- The author has inserted a mock-up or actual screenshot of the form, with annotations where visual UI enhancements are needed. Forms should be specified using hi-fi mock-ups which show all of the required elements with annotations and explanations.
- The process for initiating the output generation is clearly specified.
- Type of output specified (i.e. to a screen for download, for Desktop printing by the user, for non-interactive direct printing).
- Existing forms similar to the proposed form have been identified as appropriate, and included as samples.
- Paper size is specified (A4, Letter, Label dimensions, etc).
- Form layout is provided with real positions and sizes of each page.

- Page format and orientation are included for each page.
- Layout is harmonized with other similar forms.
- The Locale of the form is specified; this will determine things such as language, date and number formats, address formats, etc.
- If applicable, pre-printed form identified (a document is included with this pre-printed at real size or the pre-printed is sent by posting).
- Maximum lengths are suitable for layout.
- Field descriptions provided for each form field that is not obvious to a suitably experienced developer.
- Standard texts to be used are defined.
- Deviations from standard SAP address format are identified.
- The form language(s) is indicated and is not a constant, specially if translations are required.
- Form layout and translation have been provided for all (additional) languages, If any.
- Special requirements (i.e. envelope, terms and conditions file, fax number generation procedure) have been clearly documented.
- The printer type is specified and further considerations are not forgotten for special printer types (laser printer, dot printer, thermal printer, etc...).
- The paper type is specified and further considerations are not forgotten for special paper types (continuous stationary, sticker).
- Desired persons signature which are supposed to be included in form on right with necessary information and desired language must be properly mentioned.
- Logo Requirements should be consistent to project approach.

HANA CDS View

- Source and target systems to be provided.
- CDS consumption layer to be provided, i.e. Direct read, oData V2/V4, Rest API to be detailed.

Integration Process (custom)

- Logical Source and Target Systems are identified.
- Link to artifact in LeanIX to be provided.
- Initiation method is specified (e.g. user action, event, incoming message, timed job, polling, etc.)
- A UML Sequence Diagram should be included showing the flow of messages, whether a push or pull pattern is used, which communication links are synchronous and which are asynchronous, etc.
- Data Mapping is provided for mappings which are not obvious to a suitably-skilled and experienced developer
- Input and output messages are provided
- Error handling requirements are specified, including alerting requirements and expected action on failure.
- Can the integration process be safely re-run without adverse effects (i.e. it is idempotent)?
- Defines the volumes of data being handled per invocation, the frequency of invocation, and expected execution times.

Integration Process (standard)

- Logical Source and Target Systems are identified.
- Link to artifact in LeanIX to be provided.
- References, including with URL, the vendor-provided integration content that is being deployed
- Explicitly calls out any modifications or adjustments needed to be applied.
- Defines the process for updating the content of the integration process whenever a vendor releases an update for it.
- Initiation method is specified (e.g. user action, event, incoming message, timed job, polling, etc.)
- A UML Sequence Diagram should be included showing the flow of messages, whether a push or pull pattern is used, which communication links are synchronous and which are asynchronous, etc.
- Error handling requirements are specified, including alerting requirements and expected action on failure.
- Defines the volumes of data being handled per invocation, the frequency of invocation, and expected execution times.

Mobile App

- A screen mock-up or actual screenshot with annotations where visual UI enhancements are needed. At minimum use a lo-fi wireframe or sketch.
- Offline requirement to be defined.
- Requirements for camera access, additional storage access or device specifics to be provided.
- Field descriptions provided for each UI field that is not obvious to a suitably experienced developer.
- Translations have been provided for all (additional) languages, If any.
- Navigation within the app to be provided.
- Exception process/handling must be defined.

Modification

- The Fiori app/transaction/program being modified has been provided.
- OSS Note that requires modification.
- SAP Support case number that led to the modification requirement.
- Business justification needs to be added.

Program

- The selection screen has been defined.
- The output layout has been defined.
- Exception process/handling must be defined.
- Translations have been provided for all (additional) languages, If any.
- Validation logic for Screen elements need to be documented.

- Program flow logic has be defined.

Report/Analytics

- Indicates preferred UI and data modelling technology (e.g. SAC, DataSphere, embedded Fiori Analytics).
- Processing Type specified (i.e. batch, online). If the report is going to be scheduled in batch specify the frequency of execution.
- Clearly document input criteria required for processing report - (navigation to other reports via drill-down, nested reports...), if applicable and also output type (download to Excel, dashboard only).
- Report layout is provided for all new/modified reports.
- Field descriptions are provided for all new/modified reports.
- Translations have been provided for all (additional) languages, If any.

System Interface

- The Logical Systems providing the interface is identified.
- The Interface Development ID is shown.
- IDs of Integration Processes interacting with this System Interface must be included.
- Link to artifact in LeanIX to be provided.
- The relevant integration pattern used by this System Interface must be specified.
- The API type has been provided. Please note that this can only be finalized with input from the Technical team.
- A UML Sequence Diagram should be included showing the flow of messages, whether a push or pull pattern is used, which communication links are synchronous and which are asynchronous, etc.
- Input and output messages are provided.
- Error handling requirements are specified, including alerting requirements and expected action on failure.
- Can the interface be invoked multiple times without adverse consequences, (i.e. it is idempotent)?
- Defines the volumes of data being handled per invocation, the frequency of invocation, and expected execution times.

User Interface

- The author has inserted screen mock-up or actual screenshot with annotations where visual UI enhancements are needed. UI developments such as Fiori apps or enhancements should at minimum use a lo-fi wireframe or sketch.
- The to be used UI technology has been provided.
- Supported screen size(s) have been specified (Desktop / Tablet / Phone).
- Field descriptions provided for each form field that is not obvious to a suitably experienced developer.
- Translations have been provided for all (additional) languages, If any.
- Navigation not just within the UI, but also to other UI apps have been provided.Workflow (custom)
- Exception process/handling must be defined.

Workflow (custom)

- A workflow diagram must be included. This can be a free-form diagram using Draw.io but must show decision steps and actions performed by workflow agents.
- Type of Workflow, Flexible/ Business WF/ BPA, to be defined.
- Escalation process must be defined.
- Reminders for unresponsive approvers must be defined.
- Dialogue step details, incl. text to be displayed and actions to be available are provided.
- Agent determination rules must be defined in detail.
- Translations have been provided for all (additional) languages, If any.

Workflow (standard)

- A workflow diagram must be included. This can be a free-form diagram using Draw.io but must show decision steps and actions performed by workflow agents with systems mapped.
- Type of Workflow, Flexible/ Business WF/ BPA, to be provided.
- Escalation process must be defined.
- Agent determination rule configuration must be defined.
- Translations have been provided for all (additional) languages, If any.

How to find the right specification template

Below table details the Scaffold Specification template needed per the Development Id's Type of Development.

How to create a specification using the template can be found here: Creating Functional Specifications

Type of Development	Template Name
Application Job (Custom)	Functional Specification - Enhancement
Conversion (Load only)	Data Conversion Specification

Conversion (Transform + Load)	Data Conversion Specification
Conversion Custom Build with Development Id	Functional Specification - Enhancement
Enhancement	Functional Specification - Enhancement
Form (Output)	Functional Specification - Form
HANA CDS View	Functional Specification - Report/Analytics
Integration Process (custom)	Functional Specification - Integration Process
Integration Process (standard)	Functional Specification - Integration Process
Mobile App	Functional Specification - User Interface
Modification	Functional Specification - Enhancement
Program	Functional Specification - Enhancement
Report/Analytics	Functional Specification - Report/Analytics
System Interface	Functional Specification - System Interface
User Interface	Functional Specification - User Interface
Workflow (custom)	Functional Specification - Workflow
Workflow (standard)	Functional Specification - Workflow