White Paper
A ChaRMing Way to Manage SAP Changes

WP0216 | November 2015

Gavin Richardson

Gavin Richardson is an SAP-certified ALM expert who specialises in SAP Solution Manager and associated 3rd party ALM tools. He is one of the founding members of Rapid ERP (www.rapid-erp.com); providing high quality, innovative SAP consulting services to clients who want to maximise the value that ALM can bring to their SAP operation. With over 50 years of combined experience, Rapid ERP provides the marketplace with the industry's most experienced and knowledgeable SAP ALM consultants.

IT departments are often under pressure to deliver change quickly and efficiently, whilst still ensuring that a robust Change Management process is followed to safeguard the critical IT systems. It is increasingly difficult to achieve the necessary throughput of change whilst continuing to operate a rigorous process.

This has become exacerbated by end-users of IT systems becoming used to the 'Amazon.com' experience outside of the world of work. Everything they need as a consumer is just a few mouse clicks away. Processes such as adding an item to a shopping basket and completing an order are incredibly intuitive, users don't need to spend hours reading training manuals, nor do they need to attend time-sapping training courses; they can just sign-on to the system and be productive with the minimum of fuss. Why shouldn't they expect these same streamlined processes and IT systems in the business world? Intuitive solutions will surely increase their productivity, aid in user adoption and ultimately, help the organization meet its goals.

Accordingly, the need to apply change to our business processes and supporting IT systems in a timely and efficient manner has become even more relevant. To quote Winston Churchill "To improve is to change; to be perfect is to change often".

Whilst having the perfect IT system and corresponding supporting business processes might be utopia, all IT system owners accept that change must and will happen. Not to do so is to fall behind in the marketplace and lose your competitive edge.

This paper describes the challenges that IT Change Management teams face in meeting the demands of ongoing change; before showing how, with the right tools and processes, we can ensure that changes to SAP centric systems are managed correctly to deliver the innovation that the business so desperately requires.

Change Management Headaches?

Our chosen platform for IT (whether this be SAP or non-SAP) needs to support rapid transformation and not be an inhibitor of innovation. Yet, doing so has become increasingly difficult. IT systems have become increasingly complex, massively interconnected and understood by only a few within the support organization. The more changes that need to be processed, the more stretched the change delivery teams become, after all, not only do they need to deliver the innovation that the business require, but it's often the same resources who have to juggle system maintenance and other activities associated with 'keeping the lights on'.

This group needs to manage lots of different types of changes:

1) Incidents, also known as 'Break / Fix' changes

2) Small, discrete, non-invasive changes, sometimes known as 'Catalogue' changes

3) Normal changes; typically bundled into a functional release and realized over the short - medium term

4) Strategic, large scale projects delivering significant change to the IT systems over longer implementation periods

At the same time each type of change has to be managed in a careful manner:

- The requirement needs to be captured
- The change needs to be assessed and approved (or rejected)
- As the change progresses through its lifecycle it should be documented
- Original supporting documentation and process information should be updated
- Test evidence should be captured
- Approval should be granted before the change is allowed to impact the production system

If we don't have the correct processes and tools in place, taking care of these aspects can be a burdensome activity; the most common problems we find are as follows:

- **Managing change is a manually intensive activity.** This is often caused by the usage of decentralized and/or non-integrated Change Management tools. In our experience, the most commonly used toolsets for managing change in IT systems are email and spreadsheets. As a result, it can be very difficult to understand a change's current status or when action needs to be taken. When you consider that 28% of an average workday week is already spent reading and answering emails, coupled with additional 19% of the same week being taken up searching and gathering information (McKinsey, 2012), it's clear to see that minimizing the amount of time spent chasing updates on change status and responding to such requests can only help productivity.

- **Lack of traceability / Providing an audit trail.** When using decentralized and non-integrated Change Management tools it can be very difficult to trace a new piece of code or configuration back to its original requirement. Understanding who approved the change, who developed it and how it was tested can be a significant challenge. However, these are key questions that need to be answered, especially when there is a requirement to satisfy audit requests.

- **Differing processes.** All too often, different teams and projects in the same organization will operate in a different manner when it comes to managing and delivering changes to IT systems. This can mean that shared resources have to learn different Change Management processes just to deliver their work.

Having a single Change Management tool and a well-defined Change Management process would clearly help address these issues. This would then allow more of our available effort to be spent on delivering the innovations that the business actually require.

Manage Change to Accelerate Innovation

As we have discussed above, Change Management processes need to be as efficient and streamlined as possible, but at the same time must be rigorous enough to ensure that the productive IT systems are not compromised or put at risk by poor quality changes. So what does it mean to efficiently manage change? The generally accepted definition from ITIL is as follows:

'The goal of the change management process is to ensure that standardized methods and procedures are used for efficient and prompt handling of all changes, in order to minimize the impact of change-related incidents upon service quality, and consequently improve the day-to-day operations of the organization.' (Wikipedia, 2015)

The standardized process will vary in its implementation between organizations, but typically all ITIL aligned Change Management processes have the following process steps in common, as shown in Figure 1: IT Change Management Process.

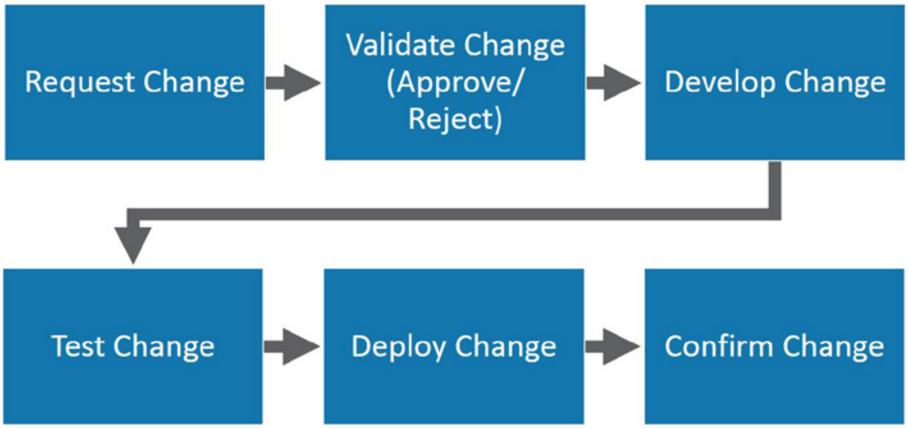


Figure 1: IT Change Management Process

The process begins with the creation of a Request for Change ("RfC"); this is created by a Requester and then updated by the Change Management team to define, as a minimum, the following information:

- Who is requesting the change?
- Who is responsible for managing the change?
- Who is responsible for developing / realizing the change?
- What is the impact of the change not being made?
- How urgent is the requirement for change?
- What priority level should the change requirement be given (ideally derived from the combined impact and urgency)?
- Any categorization to support searching and reporting
- Description of the change requirements – what needs to be changed?

The RfC must contain enough information to allow the Change Management team to make a decision on the approval (or not) of the change; this is known as the ‘Validation’ phase.

Following approval to proceed, the change is developed. This could cover a multitude of different activities, for example a change to the configuration of the system, development of new or a change to existing program code, modification of a parameter, etc. Ultimately, the requirement, as documented in the RfC, is met before the change progresses to the next phase in its lifecycle: Testing.

An important phase in any Change Management process is to ensure that all changes are tested; not just to ensure that they are fit for purpose and meet the defined requirements, but also to ensure that they don’t have any adverse impact on other parts of the system. Only once the change has been tested successfully should it be deployed to the production environment and be allowed to impact the live system.

The final stage in any changes lifecycle is its closure. The initiator of the RfC should confirm that the change has had the desired impact and then confirm or close the RfC in the Change Management system.

By using a defined process all parties involved in the Change Management process will understand the full lifecycle of a change and their role in this process; when coupled with an advanced IT Change Management system (as we will discuss in the next section) they will be much better prepared to execute IT system changes efficiently.

Once the process for recording and delivering changes is embedded, we can further enhance the process by adopting a Release Management concept. This is best described as delivering a defined and consolidated package of changes (these could be a mixture of major or minor changes, support patches, etc.) into your Test and Production systems together to yield efficiencies during the testing phase and promote integration testing of a full batch of changes, rather than each discrete change on an individual basis. Generally, organizations will use two types of release:

- **Major Release:** These cater for large scale implementations of new functionality, perhaps even new systems, usually requiring a release window of a number of months. Due to their size, they are generally run as separate projects with dedicated delivery resources. Testing is a significant component of such releases, with a requirement for unit testing of the individual changes, an integration test, user acceptance test, and, a full regression test due to the invasive nature of the changes.

- **Minor Releases:** Used for Business as Usual (BAU) operations; these minor releases deliver small, discrete changes or fixes to the production systems, typically occurring either weekly or monthly. Testing is limited, due to the overall release window, with the emphasis being on successful unit testing of the individual changes contained within the release.

Of course, outside of these release types, it is important that the change management team can also provide the flexibility to deliver Urgent or Emergency changes and fixes outside of a release where the business has a strong justification for doing so, typically where productive operation is severely impacted.

Working with a release management concept can bring significant benefits:

- **Improved stability.** Changes are promoted less frequently to the production environment, and only after all the changes within the release have been tested
- **Cost reduction.** A single regression test can cater for all changes within the release rather than having to be performed multiple times
- **Reduced administration efforts.** The release will move between test environments and into production as a single package of work
- **Improved visibility.** Changes fall into a release which has a defined delivery date for go-live, therefore the business will know when to expect their change thus helping to manage their expectations

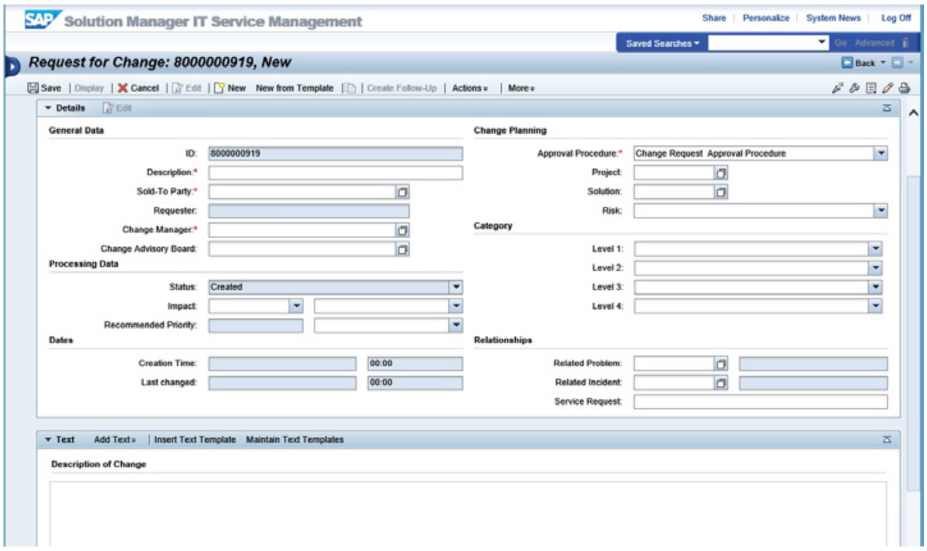
A ChaRMing Solution

In SAP centric environments there are a multitude of software solutions to choose from that can help resolve the ‘headaches’ discussed above and accelerate our ability to deliver change. However, only a few integrate completely with the SAP Transport Management System (STMS), the mechanism for delivering (or ‘transporting’) changes across the SAP landscape, and only one of these is available free of charge as part of a customer’s SAP maintenance agreement: ChaRM.

ChaRM (abbreviation of ‘Change Request Management’) is one of the core components of the SAP Solution Manager platform and once configured (which itself, does not need to be an onerous activity) goes a long way to address all of the issues and requirements we have discussed.

The process described in Figure 1 above, is supported ‘out-of-the-box’ and can be enhanced further to ensure that the main players in the process receive an email notification as soon as they are required to take action in the system. This automatically keeps the Change Management team, Developers, Testers and the business up to date with progress.

The functionality itself is user friendly, with simple screens being used to capture the details of the Change as it progresses through its lifecycle. Figure 2 shows the simple and intuitive web based ‘Request for Change’ screen, which is far more in keeping with modern user interfaces than traditional SAP screens.


SAP Solution Manager IT Service Management
Share | Personalize | System News | Log Off
Saved Searches
Go Advanced
Back
Save | Display | Cancel | Edit | New New from Template | Create Follow-Up | Actions | More
Details Edit
Text Add Text | Insert Text Template Maintain Text Templates

Request for Change: 8000000919, New

General Data

ID: 8000000919
Description:*
Sold-To Party:*
Requester:
Change Manager:*
Change Advisory Board:

Processing Data

Status: Created
Impact:
Recommended Priority:

Dates

Creation Time: 00:00
Last changed: 00:00

Change Planning

Approval Procedure:* Change Request Approval Procedure
Project:
Solution:
Risk:

Category

Level 1:
Level 2:
Level 3:
Level 4:

Relationships

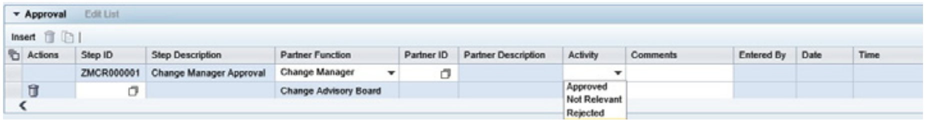
Related Problem:
Related Incident:
Service Request:

Description of Change

Figure 2: Create ‘Request for Change’ screen

As you might imagine, the ‘Request for Change’ (“RfC”) can capture all of the core information required to initiate the change approval process, such as: Requester, Description, Change Manager, Impact, Urgency, Priority, Categorization and much more besides.

A major benefit of using a Change Management systems is the ability to record the approval of the change against the originating request. ChaRM supports this approach through the appropriately named ‘Approval Procedure’, as shown in Figure 3.

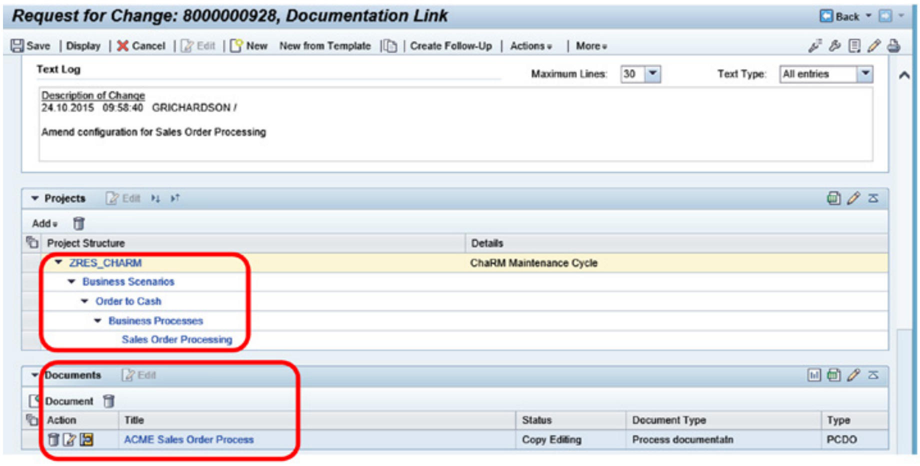


Actions	Step ID	Step Description	Partner Function	Partner ID	Partner Description	Activity	Comments	Entered By	Date	Time
	ZMCR000001	Change Manager Approval	Change Manager							
			Change Advisory Board			Approved Not Relevant Rejected				

Figure 3: Approval Procedure

The ChaRM Approval Procedure can be configured to support one or more ‘approval steps’ to enable multiple approvers for a given change; for example, you might require different levels of approval based on the type of technical change or anticipated cost. The approver of the change is recorded, alongside the date and time of the approval thus providing the necessary audit trail.

In the scenario where the IT organization has also adopted SAP Solution Manager as a central repository of process documentation (which can be connected through to BPMN toolsets to provide a graphical representation of process flow) the RfC can be linked to the associated business processes, therefore providing additional visibility of change impact, as shown in Figure 4.



Request for Change: 8000000928, Documentation Link

Back
Save | Display | Cancel | Edit | New New from Template | Create Follow-Up | Actions | More

Text Log — Maximum Lines: 30 — Text Type: All entries

Description of Change
24.10.2015 09:58:40 GRICHARDSON /

Amend configuration for Sales Order Processing

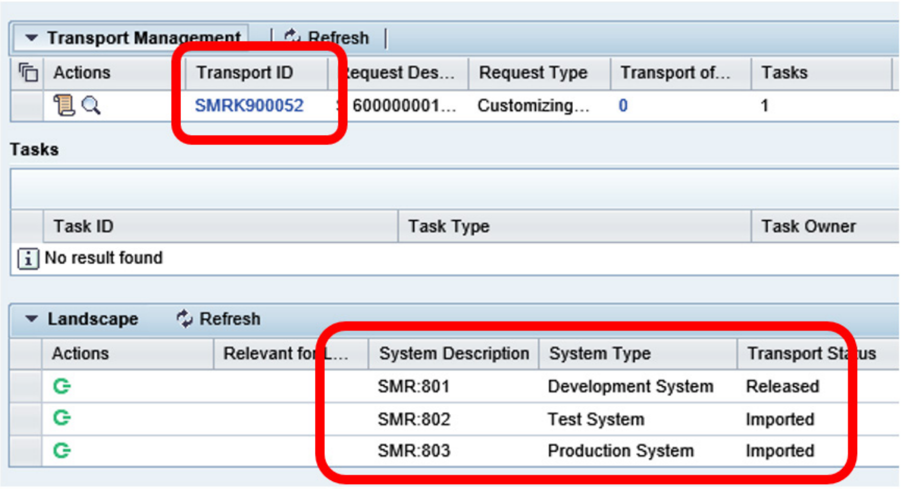
Project Structure	Details
ZRES_CHARM	ChaRM Maintenance Cycle
Business Scenarios	
Order to Cash	
Business Processes	
Sales Order Processing	

Action	Title	Status	Document Type	Type
	ACME Sales Order Process	Copy Editing	Process documentatn	PCDO

Figure 4: Process Documentation Links

Following approval, the change is then passed over to the development team to work on. The system can be configured to automatically notify the responsible developer thus ensuring that no delays are introduced into the change delivery process through a lack of communication. A vast improvement over the traditional email and spreadsheet approach.

One of the biggest benefits of ChaRM is its integration with the SAP Transport Management System (STMS). This allows for the SAP Transport(s) - the technical package used to transfer the resulting program code or configuration across the SAP system landscape – to be created and managed centrally from the Change Management system. This easily allows any person involved in the Change Management process to identify where the change has been deployed and its current status. This is shown in Figure 5.



Actions	Transport ID	Request Des…	Request Type	Transport of…	Tasks
	SMRK900052	600000001…	Customizing…	0	1

Tasks

Task ID	Task Type	Task Owner
No result found		

Actions	Relevant for L…	System Description	System Type	Transport Status
		SMR:801	Development System	Released
		SMR:802	Test System	Imported
		SMR:803	Production System	Imported

Figure 5: SAP Transports controlled via ChaRM

All of the major events are captured for the Change as it progresses through its lifecycle e.g. Test Sign Off, Production Approval (depending on change type) and Closure. This significantly improves the traceability of the change i.e. we can link the original Requirement though to Approval, Development (including SAP Transports), Test sign-off, promotion to Quality Assurance and Production systems; all from one centralized Change Management system.

Conclusion

ChaRM offers significantly more functionality than can practically be covered in this paper, for example Urgent Change support, Integration with Change Impact Analysis functions, Transport Management, and much more. However, the intent of this paper is not to go into the detail of all of the functional capabilities; rather it is to show that by making a move to a centralized, integrated Change Management tool with defined processes, accountability and auditability, we can go a long way to address the issues that were causing IT to be seen as an inhibitor to change and realize other significant benefits:

- **A Truly Integrated Change Management Solution.** SAP Transports are centrally managed from SAP Solution Manager and specifically ChaRM, ensuring a consistent approach for all SAP technologies (ABAP and JAVA).
- **Landscape Protection.** Development is controlled; no change can commence without an approved request; no untested changes can enter the production environment.
- **Improved Transparency.** Implementations can be traced back to their original requirements, including providing visibility of who approved, developed and tested any given change.
- **Ensure Validity of Process Documentation.** By linking the RFC to the stored process documentation we can help ensure that this updated as part of Change Management process and keep our repository of information relevant.
- **Flexibility.** Retain the ability to deliver change quickly without being hindered by an onerous change process.

Through adopting this Change Management toolset in SAP centric environments, we can vastly improve our ability to deliver innovation to our businesses IT systems in a controlled fashion, whilst at the same time accelerating change delivery above and beyond that possible through traditional, off system approaches using tools such as spreadsheets and emails.

References

McKinsey. (2012). The social economy: Unlocking value and productivity through social technologies [online]. Available:

www.mckinsey.com/insights/high_tech_telecoms_internet/the_social_economy

Last accessed 20th Oct 2015.

Wikipedia. (2015). ITIL [online]. Available: en.wikipedia.org/wiki/Change_management_(ITSM)#Process_overview

Last accessed 20th Oct 2015.

Orbus Software UK
London

Orbus Software US
New York

Orbus Software AUS
Sydney

Orbus Software RSA
Johannesburg

enquiries@orbussoftware.com | **www.orbussoftware.com**

Seattle Software Ltd. Victoria House, 50-58 Victoria Road, Farnborough, Hampshire, GU14 7PG. T/A Orbus Software. Registered in England and Wales 5196435