

Unterstützung der Softwareentwicklungsprozesse und Softwareprojektrealisierungsprozesse durch ein Application Lifecycle Management Tool bei Swisslog

Philippe Meuli

WDS
Swisslog AG
Webereiweg 33
CH-5033 Buchs / Aarau, Schweiz
philippe.meuli@swisslog.com

Abstract: Im Softwareentwicklungsprozess geht es darum, bei höherer Realisierungsqualität (z.B. weniger unentdeckte, ausgelieferte Fehler) und gleichzeitiger interaktiver Einbindung des Kunden zum Festigen einer konvergenten Erwartungshaltung eine grössere Flexibilität zu erreichen und dabei gleichzeitig die Entwicklungsprojektrisiken zu minimieren. Diese Anforderungen sprechen für einen agilen Ansatz den wir bei Swisslog für die Produkteentwicklung in ausgewogenem Masse unter Berücksichtigung der CMMI Praktiken diszipliniert haben. Damit stellen wir kurze Entwicklungszeiten zu einem verlässlichen Produkt und disziplinierte Abwicklung von Software Realisierungsprojekten sicher. Die Anforderungen an die Entwickler und Projektingenieure sind hoch. Sie müssen nicht nur aus teils mangelhaften Lastenheften gute Spezifikationen und Designs erstellen und daraus gute Produkte und Projekte realisieren und gleichzeitig kohärente Versionsverwaltung, durchgängige Nachvollziehbarkeit sicherstellen. Dies ist ohne integriertes Tool nur mit sehr viel administrativem Aufwand und kaum durchzuhalten-der Disziplin eines jeden Einzelnen zu realisieren.

1 Swisslog - Einleitung

Swisslog hat eine lange und erfolgreiche Geschichte in der Planung und Erstellung von Logistiklösungen als Generalunternehmer. Eine der Kernkompetenzen sind die eigenen Software- und Automationsprodukte, die es ermöglichen, Lagerbewirtschaftung, Materialflusststrategien, Materialbewegungen bis hin zu komplexen Kundenanforderungen auf professionelle Art zu realisieren.

Swisslog, ISO 9001:2000 zertifiziert, verfügt über gut eingeführte Prozesse und abgestimmte Vorgehensmodelle. Um den stetig wachsenden Anforderungen nach flexibleren Implementierungen, kürzeren Releasezyklen, zunehmendem Kostendruck und Konkur-

renz aus aufstrebenden Märkten gerecht zu werden, hat sich Swisslog entschieden einen entscheidenden Schritt in der nachhaltigen Erhöhung der Softwarequalität und gleichzeitiger Kostensenkung zu unternehmen. Einerseits geschieht dies durch ein auf CMMI (Capability Maturity Model Integrated) basierendes weltweites Software Prozessverbesserungsprogramm und andererseits durch das Offshoring von Teilen der Softwareentwicklung und Projektrealisierung in unsere Vertretung in Malaysia.

2 Prozessunterstützung und Verbesserung

Was bedeutet nun Softwareentwicklung, Softwareprojektrealisierung und Prozessverbesserungsprogramm unter einen Hut zu bringen und wie und mit welchem Tool kann dies optimal unterstützt werden? Wie erlebt Swisslog dies?

Ein Softwareverbesserungsprogramm ist vor allem ein Veränderungsprozess, den es gut vorzubereiten und zu planen gilt. Zum guten Gelingen gehören vollste Unterstützung der Geschäftsleitung und der Linienführung, uneingeschränkter Wille eines jeden Verantwortlichen die Veränderung auch selbst zu wollen, das Verständnis Änderungen als Prozess zu akzeptieren, der Wille offen zu informieren über Erfolge und gleichermassen über Misserfolge. Es handelt sich hier um einen Prozess durch und mit Menschen aus verschiedenen Kulturen und Sprachen mit verschiedenen Erwartungshaltungen und Empfindungen. Überaus wichtig sind daher die offene Kommunikation, der Gedankenaustausch und die Motivation jedes Einzelnen. Änderungen wollen erfasst und verstanden werden, Entscheidungsgrundlagen sollten transparent kommuniziert werden, Konsequenzen dürfen nicht vertuscht werden.

Produktentwicklung, Projektrealisierung über kulturelle und sprachliche Grenzen hinaus zu standardisieren und gleichzeitig mit Prozessveränderungen zu aggregieren bedingt ein wohldefiniertes Vorgehen, eine akkurate Toolunterstützung und es muss genügend Zeit eingeräumt werden. Erreichbare Zwischenziele müssen gesetzt werden und das ultimative oft unrealistische Zeitdiktat muss der gemeinsamen Anstrengung weichen, die Prozessveränderung von innen heraus gemeinsam anzupacken.

3 Vom Prozess zum Werkzeug

Ein Tool darf nicht zur Hypothek werden, sondern muss, speziell in unserem Kontext, eine länderübergreifende Unterstützung darstellen, die Abläufe vereinfacht, Nachvollziehbarkeit sicherstellt, Versionskontrolle zur Selbstverständlichkeit macht, mit den Anforderungen wächst, internationalen Standards genügt, bedienerfreundlich und zu vernünftigen Kosten zu haben ist. Wir haben uns deshalb, nach langem und reifen Evaluationsprozess mit Einbindung aller relevanten Gruppen für ein integriertes Application Lifecycle Tool basierend auf einer etablierten Versionsverwaltungssoftware entschieden. Wir sind bislang sehr zufrieden, die Benutzer verwenden das Tool aus Überzeugung und nicht aus Pflicht. Wir verwenden das Werkzeug in der Entwicklung zum Erfassen von Fehlern, Erweiterungswünschen und Anforderungen einerseits, als Führung durch den

ganzen Entwicklungsprozess andererseits. Der Entwicklungsprozess wurde abgebildet (Workflow), die Anforderungen in Funktionsmodule und einzelne Arbeitspakete (tasks) aufgegliedert und diese wiederum einzeln geplant und den Ingenieuren zur Entwicklung zugewiesen. Der Fortschritt kann auf einfache Art und Weise ermittelt und dargestellt werden.

Alle relevanten Dokumente sind somit mit dem Arbeitspaket verlinkt. Dieses Datenpaket wird, unterstützt durch das Tool, von der Anforderung über Spezifikation, Design, Codierung, Test bis hin zum fertigen Release in beide Richtungen rückverfolgbar. Somit lassen sich Einflüsse von Fehlern und Änderungen leicht ermitteln und der Aufwand abschätzen, zudem können beeinflusste Module gezielt nachgetestet werden. Die Konfigurationsmanagementfunktionalität ermöglicht es uns am Ende einer Iteration Momentanaufnahmen aller zu einem Datenpaket zugehörigen Konfigurationsobjekte zu machen (Baseline). Diese können zu jedem beliebigen Zeitpunkt wieder aufgerufen werden. Eine Baseline kann beliebige Artefakte enthalten, so z.B. Vertragsdokumente, Aufwandschätzungen, Datenblätter, Diagramme, Quellcode, ausführbare Dateien, usw. Die Versionskontrolle eines jeden einzelnen Artefaktes wird vom Tool sichergestellt. Eine Log-Datei gibt Auskunft über die Geschichte des Dokumentes, ermöglicht es die Unterschiede von Versionen darzustellen und jede Dokumentenversion wieder aufrufen zu können. Dadurch ist sichergestellt, dass die Dokumente nicht mehrfach abgelegt werden, dass immer die neuste Version als Arbeitsversion zur Verfügung steht und dass alle Änderungen nachvollziehbar dokumentiert sind.

Ein wesentlicher Vorteil liegt darin, dass unsere dezentrale, weltweit verteilte Organisationsstruktur mit zwei Entwicklungszentren und neun Software Projektrealisierungszentren durch die Engineeringumgebung voll unterstützt wird. So greifen alle Einheiten auf die zentrale Serverstruktur, die Projekte setzen je eine Entwicklungsumgebung auf die für alle Projektmitarbeiter zugänglich ist. Das bringt den Vorteil, dass alle Module in ihrer vorgesehenen Umgebung eingchecked und getestet werden und so kosten- und zeitintensive Integrationstests wesentlich effektiver abgewickelt werden können. Im Projektgeschäft bringt die Tracker-Funktionalität den Vorteil, dass Mitarbeiter unabhängig ihrer momentanen geographischen Arbeitsstätte via Web immer Zugriff haben und so Änderungsanträge, Fehlermeldungen, usw. absetzen, Code ein- und auschecken können und Zugriff auf die Offene Punkte Liste haben. Der Projektleiter kann ebenfalls unabhängig von seiner geographischen Lage z.B. in einem Flughafen, Anträge und offene Punkte bearbeiten, Planungen anpassen, Spezifikationen revidieren, Aufträge zuweisen und den Fortschritt überprüfen. Diese Aktionen sind immer für alle sichtbar.

3.1 Application Lifecycle Management Tool

Polarion, das durch uns gewählte Application Lifecycle Management Tool basiert auf Subversion und baut auf Eclipse. Es integriert die verschiedenen Funktionen einer Engineeringumgebung in einem Werkzeug und bietet gleichzeitig über den ganzen Prozess Transparenz, Nachvollziehbarkeit und raschen Zugriff auf sicher versionsverwaltete Projekt- und Prozessartefakte. Es ist gleichermassen Hilfe für Projektingenieure, Projektleiter, Führungskader und Qualitätsleiter und deckt somit ein breites Spektrum von Nutzern ab. Projektingenieure und Entwickler schätzen Versionskontrolle und Konfigurati-

onsmanagement sowie die einfache Handhabung der einzelnen Arbeitspakete. Projektleiter nutzen Planungs- und Kontrollinstrumente.

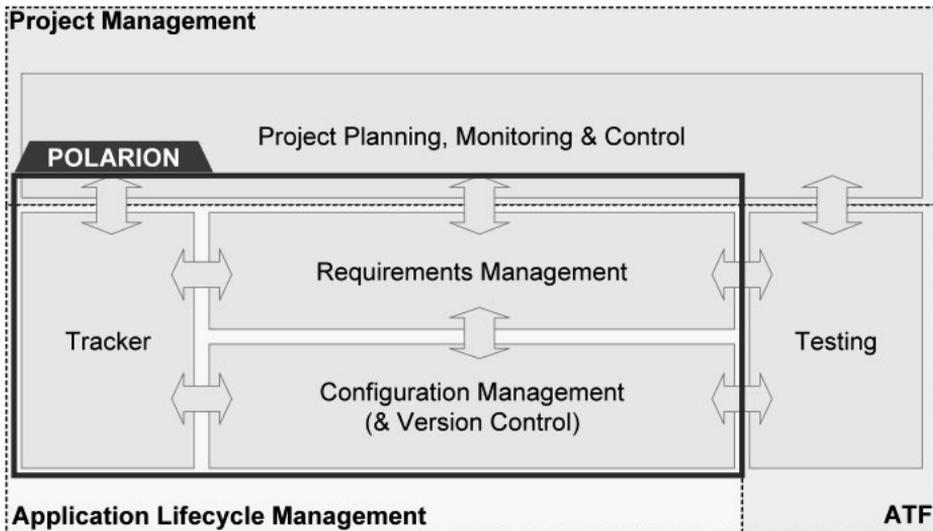


Abbildung 1 Funktionsumfang

Testartefakte werden gleichermassen verwaltet und ermöglichen so eine klare Schnittstelle zu unserer ATF (Automated Test Framework) – Testumgebung.

4 Fazit

Auf- und Ab's, Frustrationen und Erfolge auf dem Weg zu einer höheren Reife als Kollektiv festigen den Zusammenhalt einer Organisation und fördern das WIR Gefühl – mit positiver Wirkung weit über das eigentliche Ziel der Prozessverbesserung hinaus.

Beachten Sie aber bitte, dass kein Tool den Erfolg garantiert, dass Sie zuerst die Prozesse und Methoden festlegen müssen und erst dann ein Tool einsetzen sollten. Der Weg zu höherer Reife, die oben genannten Punkte berücksichtigt, ist bei richtiger Wahl weit weniger steinig als ohne Werkzeug. Bei uns entwickelt das Tool sogar eine katalytische Wirkung und sichert so den weiteren Weg des Verbesserungsprozesses.

Literaturverzeichnis

- [W01] Wallmüller, E.: SPI Software Process Improvement mit CMMI, PSP/TSP und ISO 15504. Hanser, München Wien, 2007.
- [K02] Kneuper, R.: CMMI Verbesserung von Softwareprozessen mit Capability Maturity Model Integration. dpunkt.verlag, Heidelberg, 2006.