



Abschlussprüfung Winter 2023/2024

Fachinformatiker (VO 2020) Fachrichtung: Anwendungsentwicklung
Dokumentation zur betrieblichen Projektarbeit

Workflow-Anpassungen

Anpassen von Workflows für das Änderungsmanagement in
Windchill

Abgabetermin: Pfaffenhofen, den 30.11.2023

Prüfungsbewerber:

Fabian Schmidberger

Am Galgenfeld 4a

86554 Pöttmes

Prüflingsnummer: 20525



Ausbildungsbetrieb:

Arsandis GmbH

Angkofen 5

85276 Pfaffenhofen an der Ilm

Ansprechpartner: Joachim Loos

Inhaltsverzeichnis

Abbildungsverzeichnis	III
Tabellenverzeichnis	III
Listings	III
Abkürzungsverzeichnis	III
1 Einleitung	1
1.1 Vorstellung der Arsandis GmbH	1
1.2 Projektumfeld	1
1.2.1 Systeme	1
1.3 Projektziel	2
1.4 Projektbegründung	2
1.5 Projektabgrenzung	2
1.6 Projektschnittstellen	2
1.7 Vorraussetzungen für das Verständnis	3
1.7.1 Workflows	3
1.7.2 Engineering Change Management, Änderungsmanagement	3
1.7.3 Engineering Change Request, Änderungsantrag	4
1.7.4 Engineering Change Notice, Änderungsmitteilung	4
1.7.5 Engineering Change Activity, Änderungsaufgabe	4
1.7.6 Promotion Request	4
2 Projektplanung	5
2.1 Projektphasen	5
2.2 Ressourcenplanung	5
2.2.1 Hardware	5
2.2.2 Entwicklungsumgebung	5
2.2.3 Testsystem	6
2.2.4 Weitere Software	6
2.2.5 Personal	6
2.3 Entwicklungsprozess	6
3 Analysephase	6
3.1 Ist-Analyse	6
3.2 Wirtschaftlichkeitsanalyse	7
3.2.1 Projektkosten	8
3.2.2 Amortisation	9
3.3 Lastenheft/Pflichtenheft	9
4 Entwurfsphase	9
4.1 Zielplattform	9

4.1.1	Windchill PDMLink	10
4.1.2	Programmiersprache	10
4.1.3	Datenbank	10
4.2	Geschäftslogik	10
4.2.1	ECA	10
4.3	Maßnahmen zur Qualitätssicherung	11
5	Implementierungsphase	11
5.0.1	Anpassen der Workflow-Templates	11
5.0.2	ECA	12
5.0.3	Aktivieren des Workflows	12
6	Einführungsphase	13
7	Abnahmephase	14
8	Dokumentation	14
9	Fazit	14
9.1	Soll-/Ist-Vergleich	14
9.2	Lessons Learned	15
	Literaturverzeichnis	16
	Eidesstattliche Erklärung	17
A	Anhang	i
A.1	Detaillierte Zeitplanung	i
A.2	Testprotokoll (Auszug)	ii
A.3	Lastenheft (Auszug)	ii
A.4	Admin-Dokumentation (Auszug)	iii
A.5	Aktivitätsdiagramm ECA-E	v
A.6	Workflow-Vergleich	vi
A.7	emptyRole Methode (Auszug)	vi

Abbildungsverzeichnis

1	Aktivitätsdiagramm ECA-E	v
2	Workflow vor der Umstrukturierung	vi
3	Workflow nach der Umstrukturierung	vi

Tabellenverzeichnis

1	Zeitplanung	5
2	Soll-/Ist-Vergleich	15

Listings

1	Klasse: EmptyRole	vi
---	-----------------------------	----

Abkürzungsverzeichnis

PLM	Product Lifecycle Management
AWS	Applied Water Systems
XGV	Xylem Global Vault
ECR	Engineering Change Request, Änderungsantrag
ECN	Engineering Change Notice, Änderungsmitteilung
ECM	Engineering Change Management, Änderungsmanagement
ECA	Engineering Change Activity, Änderungsaufgabe
PR	Promotion Request
VM	Virtuelle Maschine

1 Einleitung

1.1 Vorstellung der Arsandis GmbH

Die Arsandis GmbH ist ein IT- und Dienstleistungsunternehmen in Angkofen, Pfaffenhofen an der Ilm. Gegründet wurde die Arsandis im Jahr 2015 und beschäftigt zur Zeit 13 Mitarbeiter.

Das Hauptgeschäftsfeld der Arsandis GmbH liegt in der Beratung von Fertigungsunternehmen im Bereich des Product Lifecycle Managements (PLM). Darüber hinaus hat das Unternehmen ein starkes Interesse an zukunftsorientierten Technologien und verfolgt aktiv die Entwicklungen in Bereichen wie Augmented Reality, Virtual Reality und Internet of Things. Diese Technologien bieten innovative Lösungen für die Industrie der Zukunft. Weitere zukunftsorientierte Technologien werden ebenfalls verfolgt, darunter zum Beispiel AR-, VR- und IoT-basierte Lösungen für die Industrie der Zukunft. Die Arsandis GmbH hat ein breites Spektrum an Kunden aus verschiedenen Branchen. Dazu gehören Unternehmen aus dem Maschinenbau, der Automobil- und Luftfahrtindustrie sowie der HighTech- und Telekommunikationsbranche.

Als offizieller Partner von PTC liegt das Hauptaugenmerk hier meist auf dem PTC-Portfolio, das ein umfangreiches Ökosystem für die Fertigungsindustrie bereitstellt.

1.2 Projektumfeld

Durchgeführt wird das Projekt von mir in den Räumen der Arsandis GmbH, bei der ich meine Ausbildung absolviere. Das Projekt wird im Auftrag der amerikanischen Firma Applied Water Systems (AWS) durchgeführt, die in der Herstellung von Wasserpumpen tätig ist. AWS ist ein Tochterunternehmen von Xylem, einem Unternehmen, das sich auf die Entwicklung und Bereitstellung von Wassertechnologien spezialisiert hat.

1.2.1 Systeme

Produktivsystem

Der Xylem Global Vault ist der Windchill-Server von Xylem und wird von PTC in einer Cloud gehostet. Der Login auf dieses System erfolgt für die User über Single-Sign-On. Die meisten Tochterfirmen von Xylem sind auf diesem Server als eigenständige Organisation eingerichtet. Für die Durchführung des Projekts hat dieser Server allerdings keine Relevanz, da die endgültige Übernahme auf das Produktivsystem nicht von uns durchgeführt wird. Wichtig ist, dass alle Änderungen, die von uns vorgenommen werden, ausschließlich die Organisationen von AWS betreffen.

Testsystem

Dieses Testsystem ist ein Klon vom Xylem Global Vault (XGV)-Produktivserver und wird von Xylem gehostet. Der Login auf dieses System erfolgt für die User ebenfalls über Single-Sign-On. Hier nimmt der Kunde die betriebseigenen Tests vor, um unsere Anpassungen zu überprüfen. Zusammen mit meinem Projektleiter werde ich die Übernahme von unserem Entwicklungssystem auf den Trainingsserver durchführen.

Entwicklungssystem

Das Entwicklungssystem wird von Arsandis gehostet und verwaltet (Hyper-V VM) und wurde im Vorfeld als Klon des Testsystems für dieses Projekts eingerichtet, damit sämtliche Entwicklungsarbeiten darin vorgenommen werden können. Statt Single-Sign-On wurden hier lokale Testuser auf einem lokalen LDAP eingerichtet. Auch das Arsandis-Interne Testing wird hier stattfinden. Die Abschlussprojektarbeit wird - bis auf des Deployment auf den Trainingsserver - vollständig auf diesem Entwicklungssystem stattfinden.

1.3 Projektziel

Das bestehende Engineering Change Management, Änderungsmanagement (ECM) von AWS soll angepasst und erweitert werden. Dem ECM-Prozess sollen neue Funktionen hinzugefügt werden, die die Robustheit des Prozesses erhöhen und die benötigte Durchlaufzeit verringern.

1.4 Projektbegründung

Es wird eine erhöhte Robustheit des aktuellen Prozesses angestrebt, indem das manuelle Erstellen von Promotion Requests (PRs) automatisiert wird, was auch zu einer Zeitersparnis führt.

1.5 Projektabgrenzung

Das Abschlussprojekt nimmt ungefähr 80 % des kompletten Projekts ein. Die restlichen 20 % werden nach Fertigstellung des Abschlussprojekts angegangen. Es ist geplant, noch einige weitere Quality-Of-Life Verbesserungen zu implementieren.

1.6 Projektschnittstellen

Das System, an dem das Projekt durchgeführt wird, interagiert mit einigen anderen System, wie zum Beispiel

- CAD-Anwendungen über den Windchill Workgroup Manager
- Microsoft Sharepoint zur Datenübertragung

Diese sind allerdings für das Projekt nicht weiter von Belang und wurden hier deshalb nur zur Vollständigkeit aufgeführt.

Bei den Nutzern, die mit dem System interagieren, handelt es sich vor allem um Ingenieure und Manager. Ingenieure nutzen das System hauptsächlich dafür, CAD-Modelle zu erstellen, zu ändern und zu verwalten. Manager hingegen genehmigen und koordinieren Unternehmensprozesse, wie z.B. das Änderungsmanagement, auf das im Kapitel 1.7.2 noch näher eingegangen wird. Da der Windchill-Server schon lange im Einsatz ist, sind alle Mitarbeiter mit dem System bestens vertraut.

1.7 Voraussetzungen für das Verständnis

Windchill bietet umfassende Funktionen zur Verwaltung von Unternehmensdaten. Im Folgenden sollen einige dieser Funktionen vorgestellt werden, damit die Inhalte des Abschlussprojekts nachvollziehbar sind.

1.7.1 Workflows

Workflows sind Objekte, mit denen man Unternehmensprozesse abbildet. Dafür stellt Windchill einen Workflow-Editor bereit, mit dem man die Unternehmensprozesse in einer praktischen Benutzeroberfläche definieren kann. In Workflows lässt sich auch direkt Code definieren, der dann zusammen mit dem Workflow ausgeführt wird. Hier kann man auch auf Klassen und Methoden innerhalb der Windchill-Codebasis verweisen. Die folgenden Konzepte wurden und werden mit Hilfe von Workflows umgesetzt.

1.7.2 Engineering Change Management, Änderungsmanagement

Das Änderungsmanagement ist ein essenzieller Bestandteil von Windchill. Damit werden „Änderungen eines Produkts vollständig definiert und kontrolliert. Dies sorgt dafür, dass Aufgaben an die Verantwortlichen in einem wiederholbaren und automatisiertem Workflow gesendet werden“ [PTC](#). Der Change Admin ist für die Koordination des Change Managements verantwortlich. In Windchill gibt es eigentlich drei verschiedene Change Administrator Rollen, vereinfacht wird in diesem Projekt aber lediglich vom Change Admin gesprochen.

In Windchill sind drei Facetten des Änderungsmanagements implementiert: Engineering Change Request, Änderungsantrag, Engineering Change Notice, Änderungsmitteilung und Engineering Change Activity, Änderungsaufgabe. Diese werden in den nächsten Abschnitten genauer erläutert.

1.7.3 Engineering Change Request, Änderungsantrag

Der Änderungsantrag ist der erste Schritt im Änderungsmanagement. Grob gesagt wird im Engineering Change Request, Änderungsantrag (ECR) geklärt, ob die Änderung umgesetzt werden kann. Der Change Admin wählt Unternehmensabteilungen aus. Diese stimmen dann darüber ab, ob die Änderung vollzogen werden soll. Nur wenn alle befragten Abteilungen für die Änderung sind, wird der Prozess mit der Engineering Change Notice, Änderungsmitteilung (ECN) fortgesetzt. Stimmt mindestens eine Abteilung dagegen, so wird der Änderungsantrag verworfen.

1.7.4 Engineering Change Notice, Änderungsmitteilung

Die Änderungsmitteilung ist der zweite Schritt im Änderungsmanagement. Hier geht es darum, dass festgelegt wird, wie die Änderung umgesetzt wird. Dafür legt der Change Admin die Abteilungen fest, die die Änderungen durchführen sollen. Außerdem werden in diesem Schritt die Aufgaben definiert, die umgesetzt werden müssen, um die Änderung zu vollziehen.

1.7.5 Engineering Change Activity, Änderungsaufgabe

Die Änderungsaufgabe ist der dritte und letzte Schritt im Änderungsmanagement. Die geplanten Änderungen werden hier umgesetzt und anschließend vom Change-Admin überprüft. Ist der Change-Admin unzufrieden mit den umgesetzten Änderungen, so kann er eine Überarbeitung von den relevanten Abteilungen anfordern. Sobald der Change-Administrator seine Zustimmung zu den Änderungen erteilt hat, markiert dies den erfolgreichen Abschluss der Änderungsaufgabe und damit auch des gesamten Change-Prozesses.

1.7.6 Promotion Request

Der Erhöhungsantrag ist nicht direkt Teil dieses Projekts. Allerdings werden neue Mechanismen im ECM-Workflow eingeführt, die das manuelle Erstellen von Erhöhungsanträgen ersetzen soll. Demzufolge wird der Erhöhungsantrag kurz erläutert.

Ein Windchill-Objekt durchläuft stets verschiedene Phasen seines Lebenszyklus. Im Moment der Erstellung befindet es sich üblicherweise im Status 'In Arbeit'. Um das Objekt in einen anderen Zustand zu überführen, wie beispielsweise 'Freigegeben', wird ein Erhöhungsantrag durchlaufen. In diesem Prozess entscheidet ein Genehmiger, ob eine Änderung des Status des Objekts angebracht ist.

2 Projektplanung

2.1 Projektphasen

Für die Umsetzung des Projekts wurden 76 Stunden angesetzt. Der Start des Projekts erfolgt am 07.11.2023 und bis zum 27.11.2023 wird es abgeschlossen sein. Der Tabelle 1 kann entnommen werden, in welche Hauptphasen das Projekt gegliedert wurde.

Projektphase	Geplante Zeit
Analysephase	12 h
Entwurfsphase	3 h
Implementierungsphase inkl. Tests	45 h
Abnahme und Einführung	11 h
Erstellen der Dokumentation	9 h
Gesamt	80 h

Tabelle 1: Zeitplanung

Eine detailliertere Zeitplanung findet sich im Anhang A.1: Detaillierte Zeitplanung auf Seite i. Im Vergleich zum Projektantrag wurde die Erstellung der Projektdokumentation noch ergänzt. Dadurch erhöht sich die Gesamtdauer des Projekts auf 80 Stunden und die für die Erstellung eines Benutzerhandbuchs geschätzte Zeit wurde auf 1 Stunde reduziert.

2.2 Ressourcenplanung

Für die Durchführung des Projekts werden folgende Ressourcen verwendet.

2.2.1 Hardware

Von meinem Büroarbeitsplatz aus wird mein Windows 10 Arbeitslaptop genutzt.

2.2.2 Entwicklungsumgebung

Im Vorfeld des Projekts wurde bereits eine virtuelle Maschine mit dem Betriebssystem Windows Server 2019 aufgesetzt. Diese ist mit 16 GB RAM ausgestattet. Auf dieser wurde Windchill in der Version 11.1 M020-CPS026 installiert. Windchill benötigt eine Oracle Datenbank der Version 19C. Windchill benötigt eine Java Runtime Environment der Version 1.8.0 Update 202. Für die Erstellung des Codes wird die IDE Eclipse auf der VM verwendet. Die User des Windchillsystems werden im Open Source LDAP Programm OpenDJ 3.0.0 verwaltet. Eine weitere Komponente, die benötigt wird, ist ein Apache 2.4 Webserver.

2.2.3 Testsystem

Das Testsystem läuft auf einem Linux-Server ohne User Interface. Für die Verbindung zur Weboberfläche und zum Linux-Server wird das VPN Programm GlobalProtect genutzt.

2.2.4 Weitere Software

Zur Erstellung der Projektdokumentation wird IntelliJ mit dem TeXiFy Plugin und die MiKTeX Distribution für LaTeX verwendet.

2.2.5 Personal

Außerdem haben mich folgende Personen bei meinem Projekt unterstützt. Der Product Owner von AWS legt die Anforderungen fest und nimmt das Projekt ab. Ein Experte für das Windchill-System von AWS hilft mir dabei, mich in die Customizations von AWS einzuarbeiten. Als Entwickler führt der Autor die Umsetzung des Projektes durch. Der Projektleiter von Arsandis überprüft die Umsetzung und den Code. Eine Anwendungsspezialistin von Arsandis unterstützt den Autor beim Implementieren und Testen. Eine Mitarbeiterin der Personalabteilung hilft beim Aufstellen der Wirtschaftlichkeitsanalyse.

2.3 Entwicklungsprozess

Die Umsetzung meines Projekts wird grundsätzlich agil ablaufen, in dem der aktuelle Status und Änderungen an den Anforderungen nach Bedarf mit den Kunden diskutiert werden. Außerdem gibt es regelmäßige Besprechungen mit dem Projektleiter, um den Status des Projekts zu besprechen, dort wird auch diskutiert, ob es noch Optimierungsbedarf bei der Implementierung gibt.

3 Analysephase

3.1 Ist-Analyse

Dieser Abschnitt erhält Auszüge aus dem Projektantrag, um eine Übersicht über die aktuelle Situation des Systems zu geben.

Der ECM besteht aus den zwei Komponenten ECR und ECN. Aktuell werden Aufgaben im ECN getrackt, dies ist so von Windchill eigentlich nicht vorgesehen. In Zukunft sollen Aufgaben deswegen in einem Engineering Change Activity, Änderungsaufgabe (ECA) abgebildet werden. Dies führt dazu, dass die konkreten Implementierungsaufgaben logisch von der Planungsphase (ECN) abgegrenzt wird, was es dem Change Admin einfacher macht, einen Überblick über die aktuelle Situation zu bekommen.

Im ECM von AWS sollen neue Abteilungen berücksichtigt werden. Dafür müssen diese dem bestehenden ECR und ECN hinzugefügt werden.

Bei der Erstellung der ECRs werden zudem Startparameter abgefragt. Das sind Boolean-Attribute, die die automatische Auswahl von Abteilungen beeinflussen. Dies ist eine Erleichterung für den Change Admin, da dann nicht jedes Mal alle Abteilungen manuell ausgefüllt werden müssen. Der ECR ist für alle Organisationen gleich. Allerdings benötigen nicht alle Organisationen alle Attribute, die auf dem ECR definiert wurden. Dafür wurde von einem anderen Softwareunternehmen eine XML-Datei entwickelt, die nicht benötigte Attribute von einer Organisation vom UI versteckt. Es sollen neue Boolean-Attribute definiert werden, die nur für die AWS Organisationen sichtbar sind. Dazu müssen diese dem ECR hinzugefügt werden und die in der XML-Datei für die anderen Organisationen versteckt werden.

Um den Change Admin bei der Auswahl der beteiligten Abteilungen zu unterstützen, wurde von einem anderen Softwareunternehmen eine benutzerdefinierte JSP Seite entwickelt, die die Abteilungen in einer Liste darstellt. Der Change Admin kann dann über Checkboxen Abteilungen für den Prozess entfernen oder hinzufügen. Dieses Pop-Up-Fenster wurde von einem anderen Softwareunternehmen in einem bereits abgeschlossenem Projekt entwickelt.

Zudem werden Abteilungen aktuell in einem eigens dafür angefertigten Pop-Up-Fenster ausgewählt. Hier kann der Change Admin über Checkboxen Abteilungen auswählen, die am Change Prozess beteiligt sein sollen. Dieses Pop-Up-Fenster wurde von einem anderen Softwareunternehmen in einem bereits abgeschlossenem Projekt entwickelt. Allerdings bringt diese Lösung einige Probleme mit sich. Zum einen verzögern Customizations, die von den OOTB-Windchill-Konfigurationen abweichen, oft größere Windchill Upgrades, da nicht garantiert ist, dass die Customization auch in zukünftigen Versionen funktioniert. Zum anderen können nur Administratoren mit Zugriff zur virtuellen Maschine Änderungen an der Struktur des Pop-Up-Fensters vornehmen (z. B. hinzufügen oder entfernen von Abteilungen), da die Customization als JSP-Datei auf dem Server liegt. Um diese beiden Probleme zu beseitigen, soll hier auf eine Lösung zurückgegriffen werden, die bereits von Haus aus in Windchill implementiert ist, nämlich den 'Set Up Participants'-Tab. Da es sich um eine OOTB-Funktion von Windchill handelt, ist es unwahrscheinlich, dass es damit Probleme bei einem Upgrade gibt. Außerdem können Windchill-Administratoren über den Workflow Template Editor Änderungen an den Abteilungen vornehmen, ohne Zugriff auf die VM zu benötigen.

Zudem sind die aktuellen Workflowprozesse sehr unstrukturiert dargestellt und sind kaum dokumentiert. Dies macht es schwieriger für Entwickler, die nicht mit den Prozessen vertraut sind, Erweiterungen oder Änderungen vorzunehmen.

3.2 Wirtschaftlichkeitsanalyse

Da die exakten Gesamtkosten des Projekts nicht offenbart wurden, gehen wir von einem empfohlenen Festpreis von 3.000 € aus. Der Anteil dieser spezifischen Projektarbeit am Gesamtprojekt wird auf 80 % geschätzt. Daher kalkulieren wir einen Umsatz von 2.400 € für die Umsetzung gemäß der hier beschriebenen Aufgabenstellung.

3.2.1 Projektkosten

Ausbildungsvergütung

Hier haben wir ein Bruttoentgelt von 1080 €.

Arbeitgeberanteil Sozialversicherung

Die Arbeitgeber-Sozialabgaben betragen 199,92 € ohne die Unfallversicherung. Für die Unfallversicherung liegen mir keine Zahlen vor, auch eine Internetrecherche half nicht weiter. Deswegen wurde mir von der Personalabteilung geraten mit 20 € pro Monat zu rechnen.

Abstimmungskosten

Der Projektleiter von Arsandis führte mehrere interne Abstimmungsgespräche mit mir zum Fortschritt des Projekts durch. Diese Gespräche hatten das Ziel sicherzustellen, dass die Qualitätsanforderungen erfüllt werden und die Umsetzung auf bewährten Methoden basiert, während die Lösung sich gut in den Gesamtentwurf des Kundensystems einfügt. Zusätzlich nahm der Projektleiter, wie in anderen Teilen des Projekts auch, an Meetings mit dem Kunden teil. Konkrete Zahlen dafür sind für mich nicht einsehbar. Es wurde mir geraten, mit einem festen Stundensatz von 100 € zu kalkulieren.

Gemeinkosten

Intern verwenden wir keinen Gemeinkostenfaktor. Ungeachtet dessen müssen wir jedoch sicherstellen, dass unsere Gemeinkosten gedeckt werden. Aus diesem Grund werde ich eine weitere Annahme treffen, um eine standardmäßige Wirtschaftlichkeitsanalyse durchzuführen, da mir die genauen Zahlen nicht zugänglich sind.

- Energiekosten: Heizung, Strom = 15 %
- Mietkosten: Büroräume = 15 %
- Ausrüstungskosten: Clients, Monitore, Server, Peripherie, Lizenzen = 10 %
- Gehälter für Marketingabteilung: 10 %
- Gehälter für Personalabteilung: 10 %

Das führt uns zu einem Gemeinkostenzuschlag von 60 %

Gesamtkosten

Hier werden die Kosten eines Arbeitstages durch das Herunterbrechen des Jahresverdiensts auf einen Tag ermittelt. [ARB](#)Für das Jahr 2023 wurden 251 Arbeitstage angenommen.

Kosten für Arbeitstage des Prüflings = $12960 \text{ €} / (251 \text{ Arbeitstage} - 24 \text{ Urlaubstage}) * 10 = 570 \text{ €}$

Kosten für Abstimmungsgespräche = $\text{Stundensatz} * \text{Anzahl Stunden} = 100 \text{ €} * 6 = 600 \text{ €}$

Gesamtkosten = $(\text{Kosten für Abstimmungsgespräche} + \text{Kosten für Arbeitstage des Prüflings}) *$
Gemeinkostenzuschlag = $(570 \text{ €} + 600 \text{ €}) * 1,60 = 1872 \text{ €}$

3.2.2 Amortisation

Der Gewinn wird errechnet indem die Gesamtkosten vom Umsatz abgezogen werden.

Gewinn = $\text{Umsatz} - \text{Gesamtkosten} = 2.400 \text{ €} - 1.872 \text{ €} = 528 \text{ €}$

3.3 Lastenheft/Pflichtenheft

Zu Beginn des Projekts hat der Kunde bereits einen Entwurf des Lastenhefts ausgearbeitet, welcher zusammen mit mir und dem Projektleiter von Arsandis weiter angepasst wurde. Dort sind alle Anforderungen des Auftraggebers für die Anpassungen der aktuellen Anwendung enthalten. Da sich die Kunden gut mit Windchill und seinen Funktionalitäten auskennen, haben sie bereits Lösungsvorschläge eingearbeitet. Darum kann man sagen, dass es sich bei diesem Lastenheft um einen Hybriden aus Lastenheft und Pflichtenheft handelt. Aus diesem Grund wurde vom Entwickler kein separates Pflichtenheft mehr angefertigt. Im Anhang A.3: Lastenheft (Auszug) auf Seite ii befindet sich ein Auszug aus dem Lastenheft.

4 Entwurfsphase

4.1 Zielplattform

Bei der Zielplattform handelt es sich um PTC Windchill PDMLink, eine web-basierte Anwendung zur Verwaltung von Unternehmensdaten. Windchill ist mit jedem modernen Browser kompatibel. Windchill besteht aus den folgenden Grundbestandteilen:

4.1.1 Windchill PDMLink

Windchill und Windchill PDMLink werden oft synonym verwendet, da PDMLink das Basispaket von Windchill darstellt. Windchill ist eine Product Lifecycle Management (PLM) Webanwendung von PTC, die das Verwalten von Unternehmensobjekten für die Herstellungsindustrie vereinfacht. Besonders Wert wird dabei auf die Verwaltung von CAD-Modellen gelegt. CAD-Modelle sind am Computer gefertigte zwei- und dreidimensionale Zeichnungen von Konstruktionsobjekten. Mit dem Windchill Workgroup Manager existiert eine Schnittstelle zwischen Windchill und vielen CAD-Anwendungen, die auf dem Markt existieren (wie z. B. Creo Parametric, CATIA oder Solidworks). Das Hauptgeschäftsfeld von PTC ist das Engineering, darunter vor allem die Automobilindustrie oder die Luft- und Raumfahrtindustrie. Allerdings bietet PTC noch eine große Zahl an Erweiterungen an, um z. B. auch die Textilindustrie optimal in Windchill integrieren zu können.

4.1.2 Programmiersprache

Da Windchill eine Java-Applikation ist, werden tiefer gehende Anpassungen durch Java-Code erfolgen. Auch der Code in den Workflow-Templates wird in Java definiert. Um die Benutzeroberfläche um neue Elemente oder Fenster zu erweitern wird auch das JSP Framework genutzt.

4.1.3 Datenbank

Unser Entwicklungsserver läuft auf einer Oracle 19C Datenbank, Windchill ist allerdings auch mit den Datenbanken von Microsoft kompatibel.

4.2 Geschäftslogik

4.2.1 ECA

Im Anhang A.5: Aktivitätsdiagramm ECA-E auf Seite v befindet sich das Aktivitätsdiagramm für den ECA-E Prozess. Das ist eine spezielle Form des ECA-Prozesses, der alle Aufgaben der Ingenieure enthält. Der Gegenpart dazu ist der ECA-NE, in dem sich alle Aufgaben für die Nicht-Ingenieure sammeln. Dieser ist ähnlich aufgebaut wie der ECA-E, deswegen soll hier nur auf einen Prozess näher eingegangen werden.

Zuerst werden die „Affected Objects“, also die Objekte, die im Änderungsprozess geändert werden sollen, vom ECR zum ECA-E kopiert. Dadurch kann vom ECA-E auf die Objekte zugegriffen werden und es können „Resulting Objects“ erzeugt werden. Mit dieser speziellen ECA Funktion kann der Ingenieur festlegen, welchen Lebenszyklusstatus das Objekt nach der Fertigstellung des Change-Prozesses erhält.

Als nächstes wird das „Product Line Attribute“ eingelesen. Damit wird bestimmt welches Team Template im nächsten Schritt geladen wird. In den Team Templates sind User bereits zu Rollen

gemappt. Dies erlaubt es uns eine Default-Vorauswahl in den Workflow zu laden. Damit müssen die Change Administratoren nicht alle User zuweisen.

Danach werden die Startparameter des ECR in den ECA-E eingelesen. Wie beim ECR beeinflussen diese auch hier die automatische Rollenauswahl. Wurde also z. B. beim Erstellen des ECR das Attribut „Drawings“ auf Nein gesetzt, so wird der User dieser Rolle automatisch im nächsten Schritt aus dem ECA-E Prozess entfernt.

Möchte der Change Admin trotzdem von der Vorauswahl abweichen, so kann er dies in der anschließenden Aufgabe „Select users“ machen.

Im Anschluss daran werden die ausgewählten Ingenieure dazu aufgefordert ihre Aufgabe abzuschließen. In dieser können sie Änderungen an den Affected Objects vornehmen.

Haben alle Ingenieure ihre Aufgaben abgeschlossen, werden die Affected Objects automatisch vom System gesperrt. Dadurch können während der Review-Phase keine Änderungen mehr am Objekt vorgenommen werden.

Abschließend kann der Change Admin nun entscheiden, ob die Aufgaben zufriedenstellend abgeschlossen wurden. Ist er mit der Arbeit einer Abteilung nicht zufrieden, so kann er diesen eine „Rework“-Task zukommen lassen.

Wenn der Change Admin zufrieden ist, so ist der ECA-E abgeschlossen und es wird mit dem ECA-NE fortgefahren, in dem der gleiche Prozess, nur mit Nicht-Ingenieuren durchlaufen wird.

4.3 Maßnahmen zur Qualitätssicherung

Das Testing erfolgt manuell durch „durchklicken“ der Workflowprozesse. Dabei erfolgen mehrere Arsandis-Interne Tests auf unserer Entwicklungsumgebung, bei dem eine Kollegin den Prüfling unterstützt. Hier wird überprüft, ob die korrekte Funktionsweise der Anpassungen gegeben ist.

5 Implementierungsphase

5.0.1 Anpassen der Workflow-Templates

Umstrukturierung der Nodes zur besseren Übersichtlichkeit

Dieser Schritt wird im Workflow-Template-Editor von Windchill durchgeführt. Exemplarisch wird hier nur die neue Struktur des ECRs gezeigt, da sich hier die größten Änderungen ergeben haben. Der ECN Workflow hat allerdings auch eine Umstrukturierung erfahren. Es wurde besonders Wert darauf gelegt, dass die Elemente in einer klaren Struktur angeordnet werden und sich Linien möglichst nicht kreuzen. Man kann deutlich erkennen, dass der Workflow nun aufgeräumter und strukturierter ist. Zudem wurde der große Block im unteren Teil von Abbildung 2 mit einem separaten Block (Department Approval Process) ersetzt.

Ein vorher-nachher Vergleich findet sich im Anhang A.6: Workflow-Vergleich auf Seite vi.

5.0.2 ECA

Die Erstellung der beiden ECA's erfolgt ebenfalls im Workflow-Template-Editor. Hier soll vor allem der Code hervorgehoben werden, der uns die folgenden Funktionen ermöglicht.

Objekte sperren, die gerade überprüft werden

Hiermit wird die Review Transition auf den aktuellen Resulting Objects getriggert:

```
1 wt.maturity.TransitionHandlerFactory.getInstance().transitionTargets(primaryBusinessObject,wt.lifecycle.
   Transition.toTransition("REVIEW"),false);
```

Rework Transition triggern:

```
1 wt.maturity.TransitionHandlerFactory.getInstance().transitionTargets(primaryBusinessObject,wt.lifecycle.
   Transition.toTransition("REWORK"),false);
```

Transitionen sind in Life Cycle Templates festgelegt und steuern, welchen Life Cycle State das Objekt annimmt. In diesem Fall wird bei der Review Transition der „Under Review“ Status und bei der Rework Transition der „In Design“ Status aktiviert (Der In Design Status erlaubt die Bearbeitung des Objekts).

Automatisches Entfernen von Usern

Das ist ein Auszug aus dem Code-Fragment, das User aus dem Prozess entfernt, wenn sie durch die initiale Attributwahl nicht für den Prozess berücksichtigt werden sollen.

```
1 if (!SafeApp) ext.xylem.change.TeamHelper.emptyRole(ca, "SAFETY APPROVALS ENGINEER");
```

Ein Auszug der emptyRole Methode befindet sich im Anhang A.7: emptyRole Methode (Auszug) auf Seite vi.

Die Variable „ca“ stellt hier den aktuellen ECA-E, „SafeApp“ stellt das Boolean-Attribut für die Safety Agency Abteilung dar. Wenn dieses Attribut also auf *false* gesetzt wird, so wird das Department (und somit auch die Rolle) nicht benötigt. Deswegen werden daraufhin alle User aus der „Safety Approvals Engineer“ Rolle entfernt.

5.0.3 Aktivieren des Workflows

Damit die beiden ECA Workflows auch wirklich Teil des Prozesses sind, müssen erst noch ein paar Konfigurationen getätigt werden.

Anlegen neuer Types

Für die beiden ECAs müssen neue Types angelegt werden. Diese werden als Subtypes des bereits bestehenden Types ECA. Damit werden automatisch Vorkonfigurationen an die beiden Subtypes vererbt.

Life Cycle Template

Mit einem Life Cycle Template kann man nicht nur die Lebenszyklusstatus eines Objekts definieren, sondern man kann damit auch Workflows starten. In dem neuen Life Cycle Template wird deswegen definiert, dass der ECA-E bzw. ECA-NE Workflow gestartet wird.

OIR

Die Object Initialization Rules legen fest was passiert, wenn ein ECA-E oder ECA-NE erstellt wird. In der OIR legen wir fest, dass das zuvor erstellte Life Cycle Template dem ECA-E bzw. ECA-NE Objekt zugewiesen wird.

Referenz im ECN

Mit diesem Stück Code wird ein ECA-E im ECN Prozess erstellt. Dafür benötigen wir den internen Namen des ECA-Engineering Types: com.fluidtechnology.globalvault.WS.ECA_Engineering. Nachdem der ECA-E abgeschlossen ist, wird das gleiche mit dem ECA-NE wiederholt.

```
1 ext.xylem.change.ChangeUtility.createECASubtype(cn, cn.getName(), "com.fluidtechnology.globalvault.WS.  
ECA_Engineering");
```

6 Einführungsphase

Das Deployment findet zusammen mit meinem Projektleiter auf einem Terminal-Only Linux-Server statt. Die Übertragung der Änderungen auf den Testserver läuft komplett manuell ab. Mithilfe des LoadFile Frameworks wurden die Types importiert. Der kompilierte Java Code und die Datei für die Rollendefinition wurde in der Windchill Codebasis abgelegt und alte Dateien gegebenenfalls überschrieben.

Der Import der Workflows, Life Cycle Templates und OIRs funktioniert über das UI - ebenfalls manuell - direkt im Windchill Server.

Die Rollen wurden für zwei Team Templates händisch ausgetauscht. Ursprünglich war für das Abschlussprojekt vorgesehen, dass dies mit einem Java Programm automatisiert für alle Team Templates der Organisation funktioniert. Durch den Zeitdruck wurde dieses Vorhaben allerdings nach hinten verschoben und zudem auch an einen anderen Mitarbeiter übergeben.

Die Übertragung auf den Produktivserver, die nicht von uns durchgeführt wird, wird automatisiert ablaufen. Dafür bereitet ein Experte von AWS ein ANT Buildscript vor.

7 Abnahmephase

Nach dem Deployment auf das Testsystem beginnt das Test-Team von AWS ihre internen Tests durchzuführen. Dabei gehen sie, ähnlich wie bei unseren Tests, manuell durch die Workflow-Prozesse durch und begutachten auch die Workflow-Templates. Die Anpassungen werden auf ihre korrekte Funktionalität überprüft. Dabei wird auch Wert darauf gelegt, ob Workflow-Prozesse, die vor unseren Änderungen gestartet wurden, auch weiterhin reibungslos funktionieren. Im Anschluss wurde uns ein Testprotokoll zur Verfügung gestellt, in dem alle Auffälligkeiten und Fehler festgehalten wurden. Diese werden nach dem Ende des Abschlussprojekts von mir ausgebessert und die Ergebnisse daraufhin in einem weiteren Meeting vorgestellt.

Da der Product Owner von AWS noch nicht gänzlich zufrieden mit unserem Ergebnis war, wurden das Projekt noch nicht offiziell abgenommen. Dies wird erst nach dem Ende des Abschlussprojekts stattfinden.

Ein Auszug des Testprotokolls findet sich im Anhang A.2: Testprotokoll (Auszug) auf Seite ii.

8 Dokumentation

Damit der Fachbereich weiß, wie man selbst eine Anpassung der organisationsspezifischen Rollen und Attributen vornimmt, wurde ein entsprechendes Handbuch angefertigt. Dieses Handbuch ist für Windchill-Administratoren gedacht, es ist also vorausgesetzt, dass der Leser sich umfassend mit dem System auskennt und auch selbständig Änderungen an z. B. Workflows vornehmen kann. Ein Auszug dazu befindet sich im Anhang.

9 Fazit

9.1 Soll-/Ist-Vergleich

Das Projektziel wurde zur Deadline nicht eingehalten. Grund dafür ist, dass sich auf der einen Seite die Anforderungen im Vergleich zum Start geändert haben. Auf der anderen Seite war die Zeit von rund 80 Stunden deutlich zu knapp bemessen für die Fertigstellung des Projekts. Darum wurde die Deadline um einige Wochen nach hinten verschoben. Das hat auch zur Folge, dass das Projektbudget aufgestockt werden musste.

Durch den Zeitdruck und eine suboptimale Kommunikation sind außerdem einige Fehler in der Implementierung passiert. Teilweise wurden Anforderungen nicht optimal umgesetzt oder anders umgesetzt, als sich der Kunde das vorgestellt hat. Die Fehler wurden von den Testern entdeckt und

dokumentiert. Das Projekt wird auch über das Abschlussprojekt hinaus forgesetzt, dabei sollen die Fehler ausgebessert werden. Trotzdem ist der Auftraggeber grundsätzlich zufrieden mit meiner Arbeit, denn es soll noch ein weiteres Projekt folgen.

Auch das programmatische Umbenennen von Rollen wurde in der geplanten Zeit nicht geschafft. Deswegen wurde diese Aufgabe an einen anderen Mitarbeiter der Arsandis ausgelagert. Dies ist aber nicht weiter tragisch, da diese Funktion erst für den Umzug auf den Produktivserver wirklich relevant ist. Für das Testen auf den Testservern wurden einige Team-Templates händisch angepasst, wodurch diese Phase des Projekts nicht beeinflusst wurde.

Im Projektantrag wurde die Erstellung eines Use-Case Diagramms erwähnt, dies wurde während des Projekts allerdings fallen gelassen, da es für das Projekt nicht notwendig erschien.

Abschließend ist der Zeitaufwand für das Anfertigen der Dokumentation unerwartet hoch ausgefallen, dafür wurde weniger Zeit in das Erstellen der Aktivitätsdiagramme investiert, damit 80 Stunden nicht überschritten werden.

Wie in Tabelle 2 zu erkennen ist, konnte die Zeitplanung nicht wie erwartet eingehalten werden.

Phase	Geplant	Tatsächlich	Differenz
Analysephase	12 h	12 h	0 h
Entwurfsphase	3 h	1 h	-2 h
Implementierungsphase inkl. Tests	45 h	46 h	+1 h
Abnahme und Einführung	11 h	11 h	0 h
Erstellen der Dokumentation	5 h	10 h	+5 h
Gesamt	76 h	80 h	+4 h

Tabelle 2: Soll-/Ist-Vergleich

9.2 Lessons Learned

Dieses Projekt war eine hervorragende Möglichkeit, das bis dahin in der Ausbildung Gelernte anzuwenden. Da dies das erste große Projekt war, in dem ich mich mit der Zeitplanung hauptsächlich selbst befassen sollte, war dies zu Beginn eine große Herausforderung. Allerdings waren die Schulstunden zum Thema Projektmanagement eine große Hilfe, um Kundenanforderungen richtig zu planen.

Zudem habe ich durch den regelmäßigen Austausch mit dem Kunden viel im Bereich der Kommunikation dazugewonnen. Es wurden zwar bereits vor dem Abschlussprojekt regelmäßige Gespräche mit langjährigen Kunden geführt, die Kunden dieses Projekts kannten wir allerdings noch nicht zuvor. Teilweise gab es Missverständnisse, die nach einem klärenden Gespräch beseitigt wurden. Dies war eine wertvolle Erfahrung, die im zukünftigen Berufsleben Gold wert ist.

Literaturverzeichnis

Arb

Wie viele Arbeitstage im Jahr 2023? https://www.arbeitstage.de/wie_viele_arbeitstage_im_jahr_2023_Berlin.htm

PTC

PTC: *What is change management? Why is it important?* <https://www.ptc.com/en/technologies/plm/engineering-change-management>

Eidesstattliche Erklärung

Ich, Fabian Schmidberger, versichere hiermit, dass ich meine **Dokumentation zur betrieblichen Projektarbeit** mit dem Thema

Workflow-Anpassungen – Anpassen von Workflows für das Änderungsmanagement in Windchill

selbständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt habe, wobei ich alle wörtlichen und sinngemäßen Zitate als solche gekennzeichnet habe. Die Arbeit wurde bisher keiner anderen Prüfungsbehörde vorgelegt und auch nicht veröffentlicht.

Pfaffenhofen, den 30.11.2023

A handwritten signature in black ink that reads "Fabian Schmidberger". The script is cursive and fluid.

FABIAN SCHMIDBERGER

A Anhang

A.1 Detaillierte Zeitplanung

Analysephase	12
Durchführung der Ist-Analyse	8
Erstellung eines Anwendungsfall-Diagramms und Ermittlung der Anwendungsfälle	1
Analyse der bestehenden JSP-Seiten und XML-Dateien zum Customizing	3
Entwurf	3
Erstellen von Aktivitätsdiagrammen zur Veranschaulichung der Workflowprozesse	3
Implementierung inkl. Tests	45
Einrichtung des Windchill-Entwicklungssystems (Import der Kundenkonfiguration)	2
Erweiterung der bestehenden JSP-Seiten und XML-Dateien zum Customizing	1
Workflowprozesse überarbeiten (übersichtlicher gestalten, Code vereinfachen)	10
Hinzufügen von neuen Abteilungen zum Change Management Prozess	1
Ersetzen der benutzerdefinierten Seite mit Windchill out-of-the-box Lösungen	2
Integration der Change Activity in den bestehenden Change Management Prozess	4
Anpassungen an der Programmcode-Logik der Startparameter	1
Programmatisches Sperren von Objekten, die sich gerade in der Überprüfungsphase befinden	10
Workflowprozesse so gestalten, dass eine Ergänzung von neuen Abteilungen einfacher funktioniert	4
Programmatisches Umbenennen von Rollen	5
Testen des kompletten Change Management Prozesses	5
Abnahme und Einführung	11
Review durch den Projektbetreuer	2
Review durch den Kunden	2
Deployment der Anwendung auf den Testserver	5
Abnahme durch den Kunden	2
Dokumentation	9
Erstellung eines Adminhandbuchs	1
Erstellung der Projektdokumentation	8
Summe	80

A.2 Testprotokoll (Auszug)

Sr. No.	Issue/Bug	Current Status	Comments by Arsandis
4	The 9th Assigned Activity found in block Non-Engineering Implementation Tasks, found in RCW CMII ECA-NE Workflow, should be renamed to "Master Planner Changes"	Resolved	
5	I was unable to find a role name containing "BIM", or "Buildingör Content". Thus am not able to add a role or person for BIM Content to the Team Template	To Discuss	According to specification the role for BIM content is Drawings Engineer There is one role for two departments. We should add another role (say "BIM Engineer") so all will be clear. Fabian can check if similar role already exists.
6	On step 1 of the change process, I see 9 Radio button selections. I expected to see 11 Radio buttons	Resolved	FHD only has 9 radio button selections as per specification (the 2 other attributes are hidden via the xml files). Only for Lowara, Lowara Hungary and Vogel the 2 extra radio button selections are visible.

A.3 Lastenheft (Auszug)

Es folgt ein Auszug aus dem Lastenheft mit Fokus auf die Anforderungen:

Engineering Change Request:

- Add new roles and tasks to workflow and custom pages.
- 'Review tasks' page will be replaced by the 'Set up participants' page. The link to Review tasks should be visible until legacy workflows are completed.
- Non-Engineering approval process for both EMEA and NA requires 1 additional approvals of Production Manager. Production Manager is a new role for Production dept.

Engineering Change Notice:

- Addition of the ECA to existing ECN workflow template (RCW CMII ECN workflow). Requirement of 2 ECA per ECR. The first ECA would constitute all implementation tasks for engineering team and the second ECA for non -engineering teams which should begin once engineering team is completed with all changes on the first ECA.
- Affected and Resulting objects to be added by engineering ECA assignees.

- Planning and scoping of the change would be done in the ECN and the final implementation or execution tasks to begin at the ECA.
- Modify the workflow and custom pages as needed in the existing workflow such that all execution tasks are moved to the ECA. (See Annex1 - markups for custom pages and Annex 2 at the end of this document.)
- Develop a mechanism to lock the objects while they are being reviewed at the Verify Engineering Implementation Task. This could be potentially done by introducing an intermediate state “Under review”. If during the review process, the reviewer determines that some rework needs to be done, the objects change to “Rework” state and return to the engineer in an unlocked state. If during review, all objects are OK, objects change to the final intended target set as defined in the ECA.
- If atleast one rework check box has been selected during the ‘Verify Implementation task’, Routing option should be defaulted to ‘Incomplete’ to avoid user error.
- Define the intended final lifecycle state of each resulting object in the ECA. The pre-set target lifecycle state could be one of the three – Pre-Production, Released or Obsolete.
- Add new implementation tasks and notification.
- Rename existing implementation tasks and notification.

A.4 Admin-Dokumentation (Auszug)

Ausschnitt aus der Admin-Dokumentation:

Hide Radio Button Selections per Org

To hide a radio button selection for an org we need the two following files located on the server:

- <Windchill>/codebase/ext/xylem/acl/attributeacl.xml
- <Windchill>/codebase/ext/xylem/acl/acvalidation.xml

Use attributeacl.xml to find the internal name of the attribute by looking for the display name.

For example we want to hide the Motor Design attribute for organization WT. So we need to look for “Motor Design” in the attributeacl.xml. The internal attribute name is MOTORDESIGN_VALIDATION.

Use acvalidation.xml to change the visibility of the attribute by adding the Organization name to the VALIDATION list.

For the Motor Design attribute we need to find the MOTORDESIGN_VALIDATION entry and add WT to the value-list

If a hidden attribute should be visible just remove the organization from the list.

A.5 Aktivitätsdiagramm ECA-E

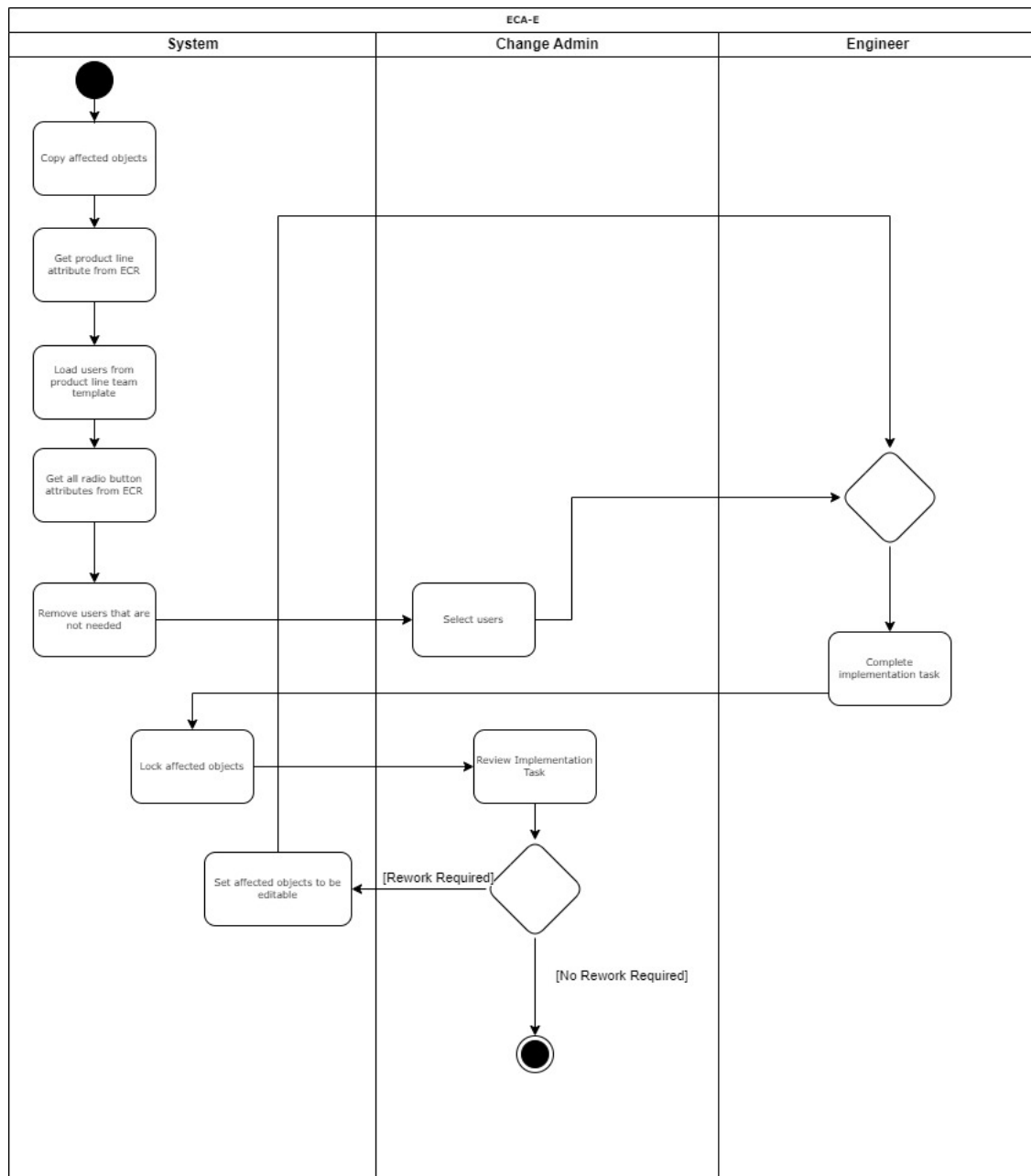


Abbildung 1: Aktivitätsdiagramm ECA-E

A.6 Workflow-Vergleich

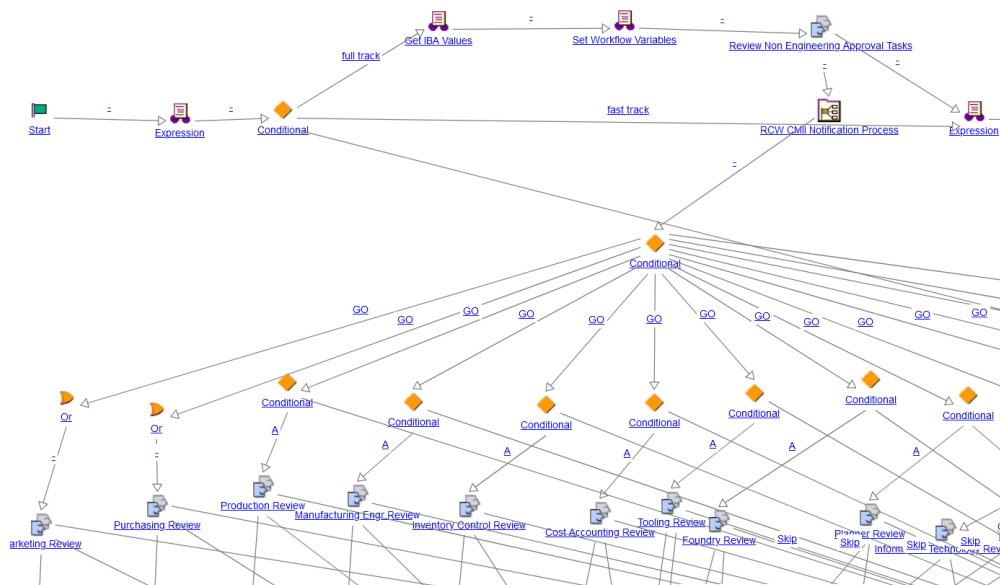


Abbildung 2: Workflow vor der Umstrukturierung

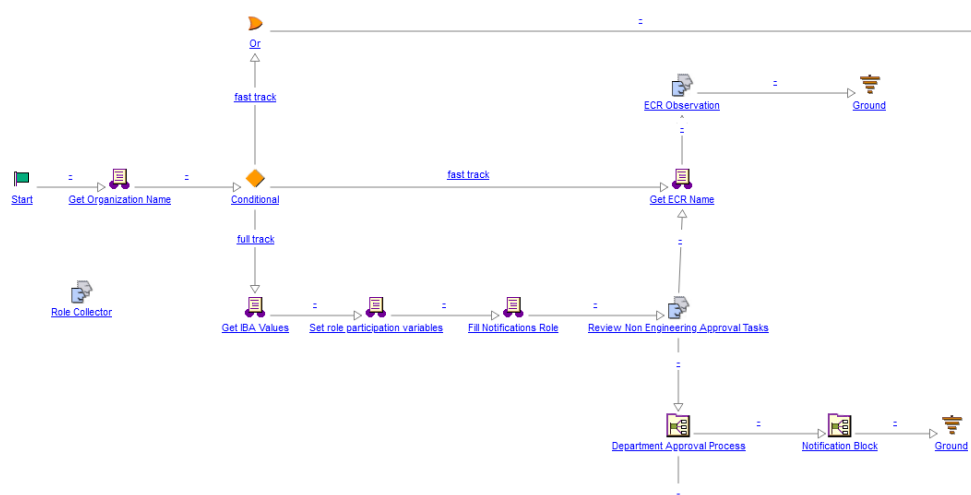


Abbildung 3: Workflow nach der Umstrukturierung

A.7 emptyRole Methode (Auszug)

```

1 public static boolean emptyRole ( TeamManaged tm, String roleID ){
2     try{
3         Role role = Role.toRole(roleID );
4         if( role == null ){
5             LOGGER.error("Did not get role for role id: " + roleID );
6             return false ;
7         }
8         TeamReference tmTeamID = tm.getTeamId();
9         if( (tmTeamID == null) ){
10            LOGGER.error( "CR team id role null: " + tmTeamID );
11            return false ;

```

```
12     }
13     Team crTeam = (Team) tmTeamID.getObject();
14     @SuppressWarnings("unchecked")
15         Enumeration<WTPPrincipalReference> crAssigneePrincipalRefs = crTeam.getPrincipalTarget(role );
16     while( crAssigneePrincipalRefs.hasMoreElements() ) {
17         WTPPrincipal existingAssigneePrincipal = crAssigneePrincipalRefs.nextElement().getPrincipal();
18         LOGGER.info("Removing principal " + existingAssigneePrincipal + " from role " + role + " in team " +
19             crTeam );
20         crTeam.deletePrincipalTarget(role, existingAssigneePrincipal);
21         crTeam = (Team)wt.fc.PersistenceHelper.manager.refresh( (Persistable) crTeam );
22     }
23     return true;
24 }catch( WTEException wte ){
25     LOGGER.error( "Error occurred while assigning role to user", wte );
26     return false ; // We may in fact have copied some roleID participants, but assume a complete failure =>
27     return false
28 }
```

Listing 1: Klasse: EmptyRole