

Software Engineering Projekt

Pflichtenheft



TECHNISCHE
UNIVERSITÄT
DARMSTADT

Ziele eines Pflichtenheftes

- Eine Festsetzung der Leistung und des Umfangs der Software
 - Anforderungen
 - Zugesicherter Funktionsumfang
 - Zugesicherter Produktumgebung
- Risikovorbeugungsmaßnahme
 - Vereinbarung zwischen Team und Auftraggeber
 - Festlegung des gelieferten Produktes
 - Festlegung eines Änderungsprotokolls während des Projektes
 - Definition zu liefernder Ergebnisse des Auftraggebers

Überblick eines Pflichtenheftes



TECHNISCHE
UNIVERSITÄT
DARMSTADT

- Beschreibung des Projekts
 - Ist-/Sollzustand; Einsatzszenarien
- Funktionale Anforderungen
 - Beschreibung der Features im Funktionsumfang
 - Grobe Priorisierung (z.B. 3 Stufig)
- Nicht-funktionale Anforderungen
- Anforderungen an Benutzungsschnittstelle
- Technische Umgebung
- Qualitätssicherung
- Deliverables
- Rechtliches
- Glossar

Beschreibung des Projekts

- Ist-Zustand
 - Die aktuelle Situation
 - Die Stakeholders, Ihre Interessen und Probleme
- Soll-Zustand
 - Vision: Ziele und Lösungsansätze, erwartete Vorteile
 - Abgrenzung des Projektumfangs: was wird gemacht, was wird nicht gemacht, Prioritäten
- Anwendungsszenarien (zum Besseren Verständnis)
 - Betriebsbedingungen: Dauer und Art der Nutzung, beaufsichtigter oder unbeaufsichtigter Server, etc.
 - Exemplarische Beschreibung komplexer Abläufe
 - Begriffsklärung komplexer Konzepte der Domäne (vgl. Glossar)

Funktionale Anforderungen

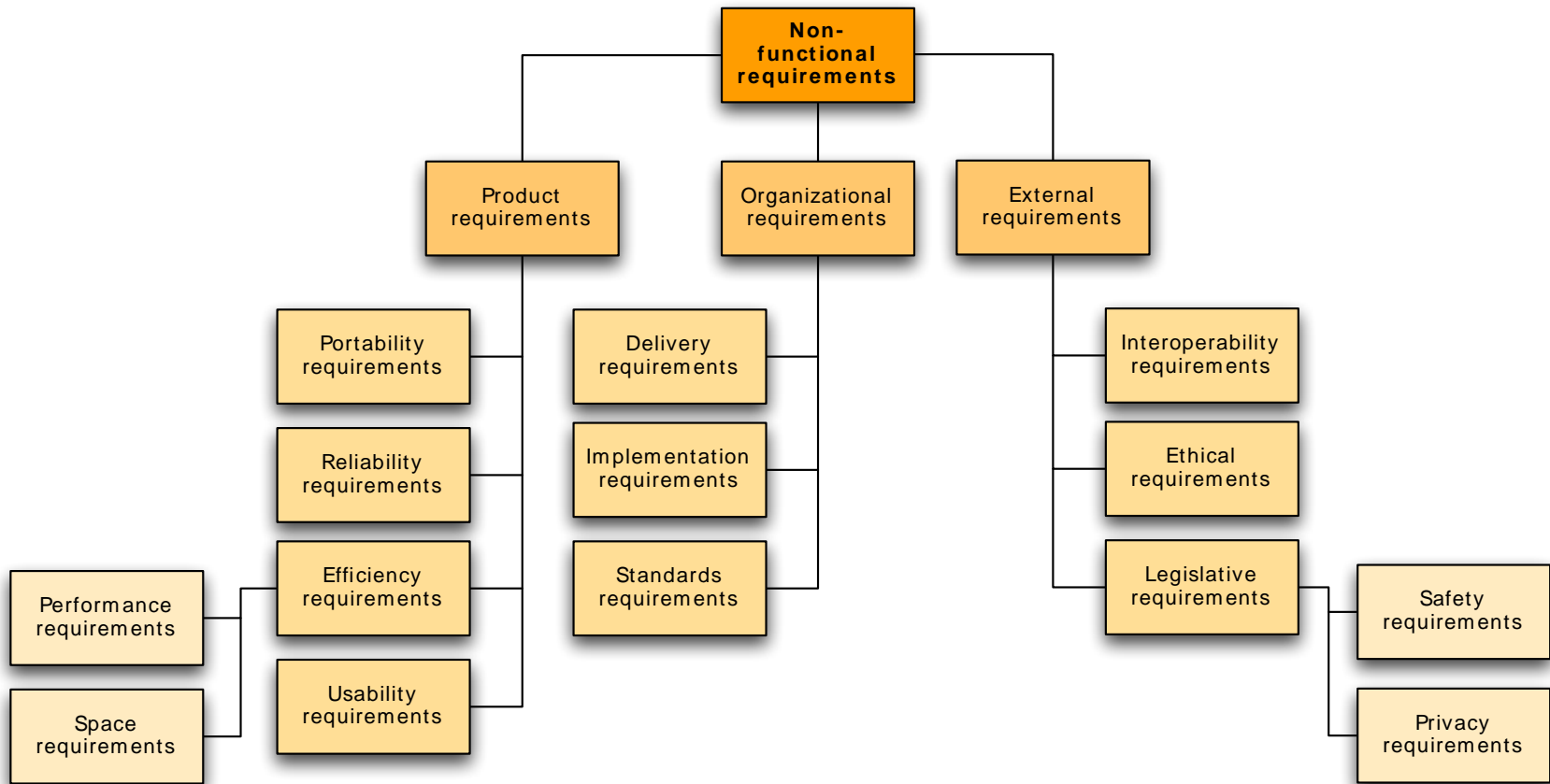


TECHNISCHE
UNIVERSITÄT
DARMSTADT

- Kernstück des Pflichtenheftes
- Überblick über alle geplanten Features der Software
- Detaillierte Beschreibung der
 - Produktfunktionen z.B. Use-Case Diagramme
 - Benutzerrollen z.B. Use-Case Diagramme
 - Datenmodel (Domänenmodel) z.B. ER-Diagramme
- Klar definierte Begriffe: vollständig; genau; verständlich
- Konsistent strukturiert und nummeriert und referenziert
- Klare Notation für Diagramme
 - Standards verwenden
 - Eigene Notation beschreiben

Nicht-funktionale Anforderungen

- Harte Anforderungen; diese müssen auch sichergestellt werden!



Beispiel NFAs

- Anforderung Unterstützung neuer Drucker
 - Sichergestellt durch Architektur der Software und Erweiterbarkeit in dieser Dimension
 - Einhaltung von Standards
- Anforderung eine Funktionalität wiederzuverwenden
 - Sichergestellt durch Kapselung der Funktionalität in Komponente
 - Schnittstelle dokumentiert

Technische Umgebung

- Produktivumgebung
- Entwicklungsumgebung
- Testumgebung

Genaue Auflistung der:

- Laufzeitumgebung mit Versionen
 - Betriebssystem
 - Virtuelle Maschine, Application-Server etc.
 - Datenbankserver, oder andere angebundenen Systeme
 - Hardware Server (falls wichtig)
- Clientumgebung (falls wichtig)
 - Betriebssystem
 - Browser (Hersteller und Versionen)
 - Zu installierende Bibliotheken

- Verwendete Produktumgebung für die Entwicklung

Falls zutreffend:

- Verwendete Versionen und PlugIns der Entwicklungsumgebung
- Spezielle Tools für spezifische Programmiersprachen
- Programmierung auf dem Host

- Welche Konfigurationen werden getestet

z.B.

Oracle + Tomcat

Oracle + Jboss
- Welche Testdaten werden verwendet
- Wie/Von wem werden Testdaten erstellt

Beschreibung der Benutzungsschnittstelle

Falls zutreffend:

- Grundlegende Anforderungen
 - Desktop-Applikation
 - Web-Schnittstelle
- Auflistung der Funktionen des UIs (im Bezug zu Funktionalität)
- Zugriffsrechte

Qualitätssicherung (feiner als in Projektplan)

- Ziele der Produktqualität (Allgemeiner als NFAs)
- Maßnahmen zur Sicherung
 - Testwerkzeuge
 - Metriken
 - Pair-Programming
 - Style-Guide etc.
- Arten von Tests (Abdeckung; Häufigkeit der Testmaßnahmen)
 - Komponententests
 - Integrationstests
 - Systemtests

- Beschreibung aller abgegebenen Artefakte (Deliverables)
- Inhalt und Form
 - Dokumentation
 - Source Code
 - Analysen
 - Installationspakete
 - ...

- Abtretung Verwertungsrechte an den Auftraggeber
- Ende des Projektes festlegen
- Organisatorische Verpflichtungen; z.B. Festlegung eines Änderungsprotokolls
- Abnahme Kriterien und Prozess
 - Art der Abnahme
 - Tests des Auftraggebers
 - Fehler Toleranz

- Beschreibung der Begriffe der Problemdomäne
- Beschreibung der Begriffe der technischen Domäne
- Nur für Zielgruppe unbekannte Begriffe
- Möglichst früh einführen und Begriffe konsistent verwenden