



Media Engineering Themenübersicht

R. Weller

- Softwarefehler
 - Murphy's Law
 - Beispiele für fehlgeschlagene Softwareprojekte
 - Definition „Scheitern“
 - Gründe für das Scheitern
- Software-Engineering
 - Definition „Software“
 - Definition „Engineering“
 - Definition „Software Engineering“
 - Historische Entwicklung
 - Softwarekrise
 - Definition „Medi Engineering“
 - Software-Development-Lifecycle

Kapitel 2: Game Engines



- Marktübersicht Medien in Deutschland
- Definition Game Engine
 - Gründe für die Verwendung von Game Engines
- Game Engines Geschichte
- Prinzipieller modularer Aufbau
 - Rendering Engine
 - Renderer
 - Szenengraph
 - Visuelle Effekte
 - GUI
 - Physik
 - Sound
 - Animation
 - KI
 - Scripting
 - Benutzerschnittstellen
 - Netzwerk
 - Ressourcenmanager
 - Core-System
- Game Loop
- Unterscheidungsmerkmale von Game Engines
 - Open Class Library
 - Extensible IDEs

- Überblick Requirements Engineering
- Definition Requirements
 - Arten von Anforderungen
 - Z.B. offen/latent, hart/weich, objektivierbar/vage
 - Funktionale vs nichtfunktionale Anforderungen
 - Qualitätskriterien
- Requirements Analysis
 - Definition
 - Ist-Analyse
 - Soll-Analyse
 - Regeln/Techniken für die Analyse
 - Richtiges Fragen
 - Vorschriften und Regeln
 - Schwierigkeiten bei der Analyse
- Definition Lasten-/Pflichtenheft
 - Definitionen Lasten-/Pflichtenheft, Anforderungsspezifikation
 - Eigenschaften eines Pflichtenhefts
 - Regeln zur Formulierung von Anforderungen
 - Begriffslexikon
 - Szenarien

- Model-View-Controller-Pattern
- Definition User Interface, HCI
- Geschichte der UIs
- Definition GUI
 - GUI-Elemente
 - Interaktionsmetaphern
- User-Centered-Design
 - Design und Prototyping
 - Skteches & Mockups
 - Storyboarding
 - Evaluation von GUI-Designs
- Regeln für das GUI-Design
 - Gestaltgesetze
 - Goldener Schnitt
 - Drittel-Regel
 - Gesetz von Fitt
 - Gesetz von Hicks
 - Magische Nummer 7
 - Poka Yoke-Prinzip
 - Regeln für Farben

- GUIs für Spiele
 - Arten von GUIs in Spielen
 - Shell-GUI, In-Game-GUIs
 - Elemente von Shell-GUIs
 - Push-Buttons, Checkbox-\ Radio-Buttons
 - Scrollbars
 - Slider
 - Listen und Drop-Down-Menüs
 - Textfelder
 - Baumstrukturen
 - Darstellung von In-Game-Informationen
 - Meta/Spatial/(Non-) Diegetic
 - Regeln für GUI-Design in Spielen

- Modellbildung
 - Definition View/Viewpoint
 - Blickwinkel auf Softwaresysteme
- Objektorientierte Analyse und Design
 - UML
 - Grundsätzliches Vorgehen
 - Akteure identifizieren
 - Use-Cases bestimmen
 - Use-Case-Diagramme
 - Statisches Modell
 - Objektdiagramm
 - Assoziationen
 - Klassendiagramm
 - Objektorientierung
 - Funktionen/Multiplizitäten/Sichtbarkeit
 - Vererbungshierarchie
 - » Liskovs Substitutionsprinzip
 - Dynamisches Modell
 - Aktivitätsdiagramm
 - Interaktionsdiagramme
 - Sequenzdiagramme
 - Kommunikationsdiagramme
 - Zustandsdiagramme
 - Randbedingungen

- Definition Design Patterns
 - Abgrenzung
 - Grundsätzliche Bestandteile
- Singleton
 - Anwendungsbeispiel
- Command-Pattern
 - Anwendung Undo/Redo
- Composite-Pattern
 - Anwendung Scripting
- Observer-Pattern
 - Variationen Pull/Push
 - Anwendung Model-View-Konzept
- Kategorien von Entwurfsmustern
 - Erzeugungsmuster
 - Strukturelle Muster
 - Verhaltensmuster

- Geschichte der Programmierertools
- Vom Quellcode zum Ausführbaren Programm
- Programmierstile
 - Namenskonventionen
 - Konventionen für Einrückungen
 - Kommentare
 - Arten und Konventionen
 - Dokumentationswerkzeuge
- Debugger
 - Funktionalität: Haltepunkte/Speicherinhalt/Call-Stack
- Profiler
- Refactoring
- Versionsverwaltung
 - Probleme bei verteilter Entwicklung
 - Beispiel Git
 - Konfliktauflösung

- Arten von Tests
 - Validierung/Verifikation
- Entwicklertests
 - Unit Tests
 - Black-Box-Tests
 - Äquivalenztest/Grenztest
 - White-Box-Tests
 - Pfadtest
 - Grey-Box-Tests
 - Integrationstests
 - Urknalle/Bottom-Up/Top-Down
 - Systemtests
 - Funktionale/Nicht-Funktionale Tests
- Freigabetests
- Benutzertests
 - Betatest/Expertenbasierte Evaluation
 - Heuristische Evaluation/Cognitive Walkthrough
 - Benutzertests/User Studies
 - Qualitative Daten (Thinking Aloud/Standardisierte Fragebögen)
 - Quantitative Daten (Eyetracker, Usability Labs,...)
 - Quantitative Auswertung der Daten
- Dokumentation von Tests
 - Testfallspezifikation
 - Testprotokoll

- Definitionen Softwareentwicklungsprozess/Prozessmodell
 - Qualitätskriterien
- Klassische Modelle
 - Code & Fix
 - Striktes Wasserfallmodell
 - V-Modell
 - Testgetriebene Entwicklung
 - Inkrementelles Modell
 - Spiralmodell
 - Rational Unified Process
 - Phasen: Konzeption/Elaboration/Konstruktion/Übergabe
- Agile Modelle
 - Agiles Manifest
 - Extreme Programming
 - Scrum
 - Rollen: Product Owner/Team/Scrum-Master
 - Product Backlog
 - Kostenschätzung
- Kriterien für Wahl des Prozessmodells

- Definition: Team
- Teambildung
 - Kommunikationsbeziehungen im Team
 - Teamrollenmodell
 - Phasen der Teamarbeit
- Kommunikation
 - Kommunikationsmodell nach Schulz und Thun
 - Kommunikationsmethoden
 - Kontrollierter Dialog/Aktives Zuhören
 - Kommunikationsprobleme
- Konfliktmanagement
 - Konfliktursachen
 - Konfliktarten
 - Verlauf von Konflikten
 - Lösungsstrategien
 - Erkennung/Diagnose/Lösung
- Motivation
 - Motivation und Gerechtigkeit
- Teamführung
 - Führungsarten (Direktiv/Mitarbeiterbezogen)

- Definitionen Projekt/Projektmanagement
- Projektinitiierung
 - Projektantrag
 - Motivation
 - Zielsetzung (SMART-Formel)
 - Nutzenanalyse
 - Projektstrukturplan (objektorientiert, ablaforientiert, funktionsorientiert)
 - Aufwandsabschätzung
 - » Heuristische/Algorithmische Methoden
 - Ablaufplanung
 - Vorgangliste/Netzplan
 - Zeitplanung
 - Meilensteine
 - Gantt-Diagramme
 - Personalplanung (Personalzuordnungsmatrix)
 - Kostenplanung
 - Projektrisiken
 - Projektdurchführungsentscheidung
 - Projektvertrag
- Controlling
 - Projektkontrolle und –Steuerung
 - Meilensteinanalyse
 - Steuerungsinstrumente
- Projektabschluss