

Zusammenfassung Web Application Engineering WS 2023

- Zusammenfassung Web Application Engineering WS 2023
 - Grundlagen Web Application Engineering
 - Definitionen Web Anwendung, Motivation
 - Definition Web Engineering
 - Definition Web Application Engineering
 - Aspekte von komplexen WA
 - Kategorien von Web Anwendungen
 - Web Service / Web Application Engineering
 - Problematiken
 - Ansätze
 - Bewährte Konzepte (WF) / Moderne Konzepte (agile Methoden)
 - Web Anwendungsentwicklung
 - Statische vs. dynamische Inhalte
 - Integration dynamischer Quellen / CGI
 - Konzepte der Web-Anwendungsentwicklung (Templating/Jessica)
 - Web Application Server
 - Moderne Web Application Server
 - Leistungsumfang von WAS wie Tomcat, PHP/Zend, Django oder Mason
 - Integration Serverseitiger Ressourcen (Sessions, DB,...)
 - Entscheidungskriterien für Tech-Stack
 - Caching
 - (Mason) Templating
 - (Mason) Staging
 - (Mason) Error Handling & Debugging
 - Software-Architekturen im Web Engineering
 - Client/Server, Multiserver Ansätze
 - Multi-Server Farmen
 - Einsatzbereich architektureller Überlegungen: Performance
 - Einsatzbereich architektureller Überlegungen: Verlässlichkeit
 - Web Application Architectures (Web Server, WAS, Software)
 - Security im Web Engineering
 - Verschlüsselungsmechanismen
 - SSL-Zertifikate
 - Sicherheit im Web
 - Security policies
 - Firewall
 - ID- & Access Management
 - Verwaltung von IAM
 - Einsatz von IAM-Systemen

- Content Management als Spezialanwendung im Web
 - CM Grundlagen
 - Online Redaktionssysteme vs CMS
 - Projektvorgehen im CM
 - Online Redaktionssysteme
 - Web Content Management System
 - Kernkomponenten von CMS
 - Kriterien für CMS
 - Content Lifecycle
 - Informationsverarbeitungsprozess
 - CM-Workflow
- Web Services
 - A2A (Application-to-Application) Kommunikation
 - Third Party Integration
- Web 2.0/Social Web, JSlibs und spez. AJAX
 - Kennzeichen von modernen Web Applikationen
 - Kommunikationstechniken

Grundlagen Web Application Engineering

Definitionen Web Anwendung, Motivation

- Softwaresystem
- Beruht auf Spezifikation des W3C
- Bereitstellung & Darstellung von Webressourcen mittels Browser
 - SW: Statische Seite keine WA
 - UI: API keine WA

Definition Web Engineering

- Anwendung von Ansätzen (Methoden, Werkzeuge)
- zur Entwurf, Implementierung, Betrieb Web-Anwendungen

Definition Web Application Engineering

- Technologische Konzepte
- Erleichterung des Informationsaustausches
- Systematische Entwicklung einer WA

Aspekte von komplexen WA

- können mehreren Kategorien zugeordnet werden
- digitalisieren traditionelle Bereiche (Online-Banking, digitales Amt)
- Integration externer Dienstleistungen (DBs, Zahlungs-API, Zugriffsrechte)
- Sicherheit (Datenschutz, Rechte)
- Zuverlässigkeit

- Performanz
 - N-tier Architektur
 - Proxying, Clustering
 - Replikation & Caching
- Management & Wartung

Kategorien von Web Anwendungen

- Dokumentenzentriert (zB: Statische Homepage)
- Interaktiv
 - dynamisch generiert durch User Interaktion
 - simpel mittels CGI (Common Gateway Interface)
 - Eingaben durch HTML Forms
 - zB: Fahrplanauskunftssystem
- Transaktional
 - Hohe Interaktivität
 - User können modifizieren
 - Nutzung von DBs
 - zB: Online-Banking, Ticketsystem
- Kollaborativ (zB: Chatrooms)
- Workflow-basiert
 - Abwicklung von Geschäftsprozessen (Workflow)
 - für Unternehmen & Behörden
 - Standardisierte Schnittstelle für strukturierte Workflows
 - Herausforderungen: Komplexität, Unternehmens-Autonomie, Flexibilität
 - zB: Patienten-Workflow im Gesundheitsbereich
- Portalorientiert (zB: Nachrichtenportal)
- Personalisiert (zB: ID- & Access-Management)
- Semantic Web

Web Service / Web Application Engineering

Problematiken

- Viel unterschiedliche SW für Design, Entwicklung & Betrieb; nicht für den gesamten Lebenszyklus
- Web-Datenformate final formatiert
 - Wenig Wiederverwendung
 - schlechte Verwaltbarkeit & Wartbarkeit

Ansätze

- Realisierung
 - HTML schreiben
 - Seiten Editoren
 - dynamische Seitenerzeugung (Scripts)

- ASP, PHP, JS
 - Server-side vs Client-side
- OO Site Engineering Tools
 - Mason, (W3Objects, JESSICA)
- Frameworks
 - Client: React, Angular, Vue
 - Server: LAMP, PHP, Mason, Ruby (on Rails)
- selbes Ergebnis: Nutzer sieht HTML
- Auswahl: Anforderungen, Abhängigkeiten, Infrastruktur, Ressourcen

Bewährte Konzepte (WF) / Moderne Konzepte (agile Methoden)

- Wasserfall
 - Requirement Analysis
 - Design
 - Implementation & Testing
 - Deployment
 - Maintenance
- Agile Methoden
 - viele kleine Iterationen
 - mehrere Releases
 - zB: Scrum, XP

Web Anwendungsentwicklung

Statische vs. dynamische Inhalte

- Statisch
 - zB: Private Homepages, klein Info-Seiten
 - Page-Editor bis simples CMS
- Dynamisch
 - begann mit CGI
 - Web Application Server (WAS)

Integration dynamischer Quellen / CGI

- dynamische Inhalte aus "Legacy" Apps
- relevante Info die in WWW eingebunden wird
- zB: Lagerbestandssystem, Bibliothekssystem, ...
- Anbindung an DB und Managementsysteme
- 3+ Tier Lösung: Client (Browser), (Web)Server (WAS), (Legacy) Application

Konzepte der Web-Anwendungsentwicklung (Templating/Jessica)

- Templates als vorlage für Dokumente
- Vorlagen aus CMS mit Daten aus DB befüllen

Web Application Server

- Definition: *Application Server*
 - Supports thin clients
 - distributed computing
 - manage client session, host business logic
- Definition: *Web Application Server*
 - Application Server per Web-Server erreichbar
- Zusammengehöriges Toolset: Framework
 - Client (React, Vue) vs. Server (LAMP)
 - LAMP: Linux, Apache, MySQL, PHP/Perl/Python
 - MEAN: MongoDB, Express, Angular, NodeJS

Moderne Web Application Server

Leistungsumfang von WAS wie Tomcat, PHP/Zend, Django oder Mason

- Web Server
 - Netzwerk Protokolle (HTTP/S, SSL)
 - statische Dokumente
 - Proxying
- WAS SW Coordinator (zB PSGI Server: starman)
 - Memory Management
 - Start Parameter & Config
 - Integration mit Web Server (zB Servlets, CGI, mod_php)
- Web Framework (Application zB: Perl/Mason)
 - Templating
 - Caching
 - Zugriff auf Web Server & WAS Coordinator

Integration Serverseitiger Ressourcen (Sessions, DB,...)

- DB-Anbindung
 - persistente DB-Verbindung
- Session Management
 - Persistente Zustände (HTTP zustandslos)
 - `$_SESSION` in PHP
 - `$m->{session}` in Perl/Mason
- User tracking & Error handling
 - Statistik über Nutzeraktivitäten
 - Fehleraufzeichnung
- Server-side caching

Entscheidungskriterien für Tech-Stack

-  Yes

- Project's size & complexity
- Targeted application extensibility
- Target platform
- Resources: Time, budget, staff
- **✗** No
 - Earlier solutions (?)
 - Personal preferences (?)
 - Rival's project
 - Online Information (?)

Caching

- Spezialform volatiler Replikation
- Einzeldaten, Komponenten und ganze Seiten
- Einsatz bei ressourcenintensiven Services
- Konsistenz
 - Nutzer soll immer neueste Version sehen
 - Multiple-write Anomalien
 - CAP (Consistency, Availability, Partition Tolerance)
 - Immer nur 2 erreichbar
 - BASE (Basically available, Soft state, Eventually consistent)
- Master/slave
 - Zugriffsrechte für schreiben/lesen

(Mason) Templating

- folgt hierarchischem Aufbau des FS
- Abarbeitung des Template-Baums beginnend mit Base.mc
- Zugriff auf die Attribute der Komponente
- "Objekt orientiert"

(Mason) Staging

- Saubere Trennung zwischen Dev- & Production
- Separate Bereiche (schnelles Auswechseln möglich)
- Automatische Wahl der richtigen Komponente

(Mason) Error Handling & Debugging

- Stacktraces (Fehlerbericht) bei Perl Fehlern
- Konfiguration eines Default-Files statt automatischer Error-Page; Nutzer-freundliche Fehlerseite

Software-Architekturen im Web Engineering

Client/Server, Multiserver Ansätze

- Client & Server: Clients invoke services from one server
- Client & Multi-Server: Multiple servers handle client requests
 - Replikation oder Partitionierung für Lastverteilung & Ausfallsicherheit
 - Gegenseitige Kommunikation & Synchronisierung

Multi-Server Farmen

- Mehrere physikalische Maschinen
- Mehrere virtuelle Maschinen
 - Container (zB: Docker Instanzen)
 - Cloud Plattformen
 - fertige Standardservices (zB Email, DB, ...)
 - Orchestration (zB Kubernetes) eigener Container

Einsatzbereich architektureller Überlegungen: Performance

- Benutzer verlangen schnelle und konsistente Antwort
- daher...
 - wenig Komponenten
 - lokale Kommunikation
 - Austausch weniger Daten
- Schwächste Komponente limitiert Durchsatz
- Lastbalancierung: Verschiebung von Arbeit auf weniger ausgelastete Komponenten

Einsatzbereich architektureller Überlegungen: Verlässlichkeit

- Korrektheit
- Sicherheit CIA (Confidentiality, Integrity, Availability)
- Fehlertoleranz

Web Application Architectures (Web Server, WAS, Software)

- Verbindungstypen
 - One time: Neue Verbindung für jede Anfrage/Ressource
 - Persistent: Verbindung wird offen gehalten -> bessere Performance
- Hardware Architektur
 - bare metal
 - 1 HW & 1 OS
 - virtueller Server
 - mehrere OS parallel auf einem physikalischem Computer
 - containerized Server
 - konsistente & isolierte Umgebung
 - portabel
 - skalierbar
 - kosteneffektiv
- Server-seitige Architektur
 - Web Server simpler & leistungsfähiger Service

- Ansätze
 - monolithisch
 - parallele Prozesse
 - load balancing
- Cluster
 - über gemeinsames Frontend angesprochen -> (Reverse-)Proxy
 - dispatch zufällig oder gesteuert
 - load balancing
 - round robin
- Container Architektur
 - Orchestration: Docker Swarm, Kubernetes
 - horizontale & vertikale Skalierbarkeit
- Applikations Architektur
- WAS Architektur
- Apache Internal Architektur
 - Modularer Aufbau
 - Cluster-Modell
 - Request wird von jeder Schicht bearbeitet

Security im Web Engineering

Verschlüsselungsmechanismen

- Definition: Reversible Verunkennlichung
- Sicherheit durch Länge des Schlüssels bestimmt
- Symmetrisch zB: AES
 - nur ein Schlüssel
 - Blöcke gleicher Länge
- Asymmetrisch zB: RSA
 - zwei Schlüssel (private & public)
 - Anwendungen
 - Verschlüsselung
 - Authentifizierung
 - Digitale Signatur & Zertifikate

SSL-Zertifikate

- Bestätigung eines vertrauenswürdigen Dritten
- Signiert von CA (certification authority)
- enthält...
 - public key des Servers
 - Name (Domain) des Servers
 - Ablaufdatum
 - Signatur der CA
- benötigt für HTTPS
 - TLS tunneling

Sicherheit im Web

- Authentifizierung
- Schutz von personenbezogenen Daten
- Schutz von Zahlungsdaten

Security policies

- Mechanische Beschränkungen
 - Daten unzugänglich
 - Zulassungsbeschränkungen
 - Benutzer-, Rechner-, Domänebene
 - Firewalls, Intranets, VPNs
- Logische Beschränkungen
 - Daten zugänglich aber nicht verwertbar
 - Verschlüsselung

Firewall

- Verhindert Zugriffe auf interne Ressourcen von außen
- Konfiguriert welcher Austausch zwischen "innen" und "außen" erlaubt ist
- Typen
 - Network Layer: Packet-Filter
 - Application Layer: Gateway
- Config
 - Black-List: Alles zulassen, außer es ist verboten
 - White-List: Alles verbieten, außer es ist zugelassen

ID- & Access Management

- Wer hat Zugriff auf was?
 - ohne IAM schwer nachvollziehbar
- Gruppierung zu Services
- Identifikation von Nutzern
 - als Gruppe
 - mittels ID-Portale
- Einzel-Service
 - ein konkreter Service
 - Nutzer benötigt bestimmte Berechtigungen
- Service Portal
 - Verbundene Services (?)

Verwaltung von IAM

- selbst
- ID-Provider
 - Authentifiziert Nutzer
 - Verwaltet Nutzer Credentials

- ID-Portal
 - Komplexe Organisationen bündeln Nutzer-Authentifizierung an einem Ort
 - SSO (Single-Sign-On)

Einsatz von IAM-Systemen

- Nachteile wenn nicht genutzt
- vereinfacht und automatisiert
- erlaubt SSO
- überall wo Nutzer Authentifizierung notwendig
- Faktoren
 - Username & Passwort
 - Biometrisch
 - Client-Gerät (2-FA, multi-FA)
- Id-Portal/Provider zB: keycloak
- Id-DB zB: LDAP

Content Management als Spezialanwendung im Web

CM Grundlagen

- Web *Service* Management
 - Architekturelle Maßnahmen
 - Betrieb zB: Auswertung von Nutzungs- und Laststatistiken
- Web *Content* Management
 - Inhaltliche Wartung
 - Datenänderung
 - Navigationsstrukturierung
 - SEO (Search Engine Optimization)
 - Optik
 - Redaktionssystem & CMS (Content Management System)
- Content
 - Trennung von Inhalt, Struktur und Darstellung
- Content Management
 - Erstellen & Verwalten von Inhalten
 - Aktualität
 - Konsistenz
- CMS
 - für Pressemitteilungen, Blogs, Veranstaltungen
 - Kommerzielle Produkte: WebCenter, Magnolia, RedDot
 - Kostenlose Produkte: WordPress, Drupal

Online Redaktionssysteme vs CMS

- Online Redaktionssysteme
 - aktuell Recherchiertes schnell online bringen

- CMS
 - Verschiedenartige Inhalte
 - Verschiedene Formate
 - Hochladen und Verwalten

Projektvorgehen im CM

- Evaluierung & Einführung eines CMS
 - viele CMS am Markt
 - bedeutet Umstellung der Workflows -> Change Management
- Ist-Analyse
 - Interviews mit Mitarbeitern
 - Wünsche und Anforderungen
 - Existierende und erwünschte Internetangebote (öffentlich) und Intranetangebote (intern)
- Soll-Analyse (Mögliche Anforderungen)
 - Dezentrale Inhaltspflege
 - Rechte & Rollen
 - Schulungsaufwand
 - Einführungsaufwand
 - Export/Import von Daten (APIs)
 - Übernahme von existierendem Angebot
 - Abbildung von Workflows

Online Redaktionssysteme

- User sieht nur Teil des Service (Rollen)
- Anwendung oft zeitkritisch
- Inhalte haben Gültigkeitsdauer
- Content Management Interface
 - Steuerung der Inhalte
 - mittels Web UI oder eigener SW
- Content Management Engine
 - mittels DB
 - Zugriffsrechte & Meta-Infos
- Content Publishing Engine
 - Informationen darstellen
 - Scripts erzeugen Layout

Web Content Management System

- Redaktionssystem/CMS
- mit Web-UI

Kernkomponenten von CMS

- Trennung von Inhalt, Struktur und Darstellung

- Workflow Komponente
 - Publikationsprozess
 - Content Life Cycle
- Asset Management
 - Editieren von Inhalten und Vorlagen
- Zugriffsverwaltung
- APIs

Kriterien für CMS

- Mandantenfähigkeit
- Versionskontrolle
- Workflows
- Kollaboration
 - zB: Messaging, Status-Reports
- Personalisierung
- Skalierbarkeit & Performance
- Verlässlichkeit
- Sicherheit
- APIs
- UI
 - Tools für Autoren und Admins
 - WebUI oder eigene App
- Templates
 - Anpassbarkeit ohne Programmierung
 - Editierrechte
- Suche und Indizierung
 - Volltext-/Strukturierte Suche
 - Klassifizierung: Kategorie, Meta-Infos, automatische Beschlagwortung
- Replikation & Integration
 - Automatische Backups
 - Synchronisierung
 - maschinenlesbare Content-Formate für APIs
- Plattformunabhängigkeit

Content Lifecycle

1. Content Generierung - Verfassen - Sammeln
2. Content Organisation - Strukturieren - Indexieren - Filtern
3. Content Aufbereitung - Verfeinern - Redaktion - Kontext erzeugen
4. Content Distribution - Publikation - Notifikation - Durchsuchbarkeit
5. Content Nutzung - Kommentieren - Visualisieren / Darstellen
6. Content Reduktion - Archivieren - Löschen

Informationsverarbeitungsprozess

1. Erstellung

2. Kontrolle - Zurück zu Erstellung bei negativer Kontrolle
3. Freigabe
4. Publikation
 - Inhalte jetzt öffentlich sichtbar
5. Archivierung
 - öffentliches oder internes Archiv

CM-Workflow

- Unterstützung mittels...
 - Automatisierung
 - Rollen/Berechtigungen
 - Benachrichtigungen
 - Freigabemechanismen
- Manuelle
 - kein Automatismus
 - Kommunikation durch Mitarbeiter
- Statisch
 - vordefinierte Workflows (4-Augenprinzip)
 - übersichtlich
- Custom
 - Modellierung und Überwachung komplexer WF
 - Workflow Management im CMS oder einlesen mittels Schnittstelle

Web Services

A2A (Application-to-Application) Kommunikation

- Kommunikation zwischen Servern (bzw. nicht Browser <-> Server)
- mittels SOAP (HTTP: XML), REST (HTTP: XML, JSON, ...), JSON-P
- Integration von Services zB: Legacy Apps, Payment services

Third Party Integration

- Zahlungsdienste
 - Kreditkarten-Acquier
 - EPS-Bankenstandard
 - Paypal
- Lookup Services
 - Validatoren: Email-Adressen, IBAN, Syntax
 - Adress-Service (Geo-Info)
- Export Service

Web 2.0/Social Web, JSlibs und spez. AJAX

Kennzeichen von modernen Web Applikationen

- Web als Plattform
- Daten-getriebene Anwendungen
- Mitwirken der Nutzer
- Kontinuierliche Weiterentwicklung
- Interaktive Oberfläche
- Dynamisch
- Erscheinen wie Desktop-App

Kommunikationstechniken

- AJAX (Asynchronous Javascript and XML)
- JS API im Browser (XMLHttpRequest, fetch)
- Austausch von Daten mit Server
 - Browser holt Daten von Server
 - kein Page-Reload
 - Austausch von Teilen der Webseite