

Expert Sessions

MIXED
MODE
an Ingenics company

EXPERT SESSIONS

technik.mensch.leidenschaft

Über uns

technik.mensch.leidenschaft

Seit 1994 unterstützen wir unsere Kunden erfolgreich im Embedded & Software Engineering. Mixed Mode beschäftigt derzeit über 110 Mitarbeiter*innen.

Ob Sie individuelle Lösungen benötigen, qualifizierte Expert*innen für Ihr Team suchen oder innovative Ideen und Technologien für Ihre Projekte benötigen – greifen Sie auf unser Know-how zurück.

Qualität und höchste Kundenzufriedenheit bilden die Basis für eine erfolgreiche und langfristige Zusammenarbeit.

Unsere Kunden sind Global Player und innovative mittelständische Unternehmen aus allen Schlüsselbranchen. Sie schätzen uns als zuverlässigen Partner gemäß unserem Motto:

Unsere Kernkompetenz ist **Der Mensch** Wir machen unseren Job mit **technik mensch leidenschaft**
steht bei **uns im Mittelpunkt**

WARUM MIXED MODE?

- Stabile Firma in privater Hand
- Flexible Projektmodelle, lösungs- und ergebnisorientiert
- Kurze Reaktionszeiten und hohe Qualität
- Technisch versierte Ansprechpartner*innen
- Über 2.000 Personenjahre Projekterfahrung seit mehr als 25 Jahren
- Stetiges und gesundes Wachstum
- Branchen- und Themenfokussierung
- Umfangreiches Partner- und Know-how-Netzwerk
- Langfristige Geschäftsbeziehungen und Partnerschaften
- Gute Kundenstruktur vom innovativen Mittelstand bis hin zum Großkonzern

Themengebiete

Embedded & Software Engineering	6
Parallelisierung und verteiltes Datenflussdesign.....	6
Asymmetric Multiprocessing auf heterogenen Multiprozessorsystemen	7
Specification by Example	8
Einführung in die Systemmodellierung mit SysML	9
Variantenmodellierung mit SysML	10
SysML-Einführung für Systemingenieur*innen	11
Effizienzsteigerung durch Scrum (Workshop)	12
Einführung in Embedded Systems (Workshop).....	13
Mikrocontroller (µC) – Erste Schritte in der Programmierung (Workshop)	14
Field Programmable Gate Array (FPGA) für Einsteiger (Workshop)	15
Digitale Signalprozessoren (DSP) – Einführung in die Welt der DSPs (Workshop)	16
Echtzeitbetriebssysteme am Beispiel FreeRTOS.....	17
Grundlagen der Echtzeitsysteme	18
Erfolgreiche Softwareprojekte durch Requirements Engineering	19
Requirements Analyse mit Use Cases	20
Brauchen agile Teams noch Führungskräfte?	23
Embedded Linux	24
Linux-Grundlagen (Workshop).....	24
Debug your Linux – Praktisches Debuggen auf echter Hardware (Workshop)	25
Application Programming Kick-Start.....	27
Embedded Security	28
Entwicklungsbegleitend Testen mit dem Mixed Mode Hardware-in-the-Loop Framework (HiLF).....	28
Security for Safety am Beispiel der IEC 62443 im Kontext zur IEC 61508.....	29
Einstieg in die TrustZone mit OP-TEE	30
Advanced Security mit Hardware Security Modulen.....	31
Embedded Security mit FPGAs – Dynamische partielle Rekonfiguration am Beispiel von Xilinx-FPGAs	32
Vom ITSM zum Sicheren Eingebetteten System.....	33
Sicherheitslösungen für ressourcenbeschränkte Mikrocontroller	34
Agile and Safety – Can this work?	35

Professional User Interface	36
GUI-Gestaltung	36
Plattformübergreifende App-Entwicklung	37
Software Test & Quality	38
Softwarecode-Metriken	38
Code-Metriken konkret: Softwarequalität erfassen	39
Legacy Code: Wie teste ich chaotische Software? (Workshop)	40
Continuous Integration (Workshop)	42
Continuous Integration: Nightly Build	44
Internet of Things	45
IoT: Technologien und Entscheidungsgrundlagen für das Internet der Dinge (Workshop).....	45
Mein IoT-Gateway – von der abstrakten Architektur zur konkreten Implementierung.....	46
Maschinensteuerung mit Web-Technologien	47

Seminar-Partner



Parallelisierung und verteiltes Datenflussdesign

ABSTRACT

Bei den verteilten Systemen beschäftigt man sich mit der Aufteilung von Aufgaben über mehrere Prozesse bzw. Rechner mit dem Ziel der Redundanz und Skalierbarkeit. Die Parallelisierung eines Problems ist dabei die Voraussetzung für eine horizontale Verteilung, daher wird Ihnen im Vortrag zunächst beispielhaft ein paralleler Algorithmus vorgestellt. Danach werden einige Überlegungen zum Datenflussdesign eines verteilten Systems mit verteilten Message-Queues (im Beispiel: Kafka) präsentiert.

Schließlich wird auf komplexere Probleme wie konsistente Snapshots und Orchestrierung eingegangen.

ZIELGRUPPE

Entwicklung

Asymmetric Multiprocessing auf heterogenen Multiprozessorsystemen

ABSTRACT

Heterogene Multiprozessorsysteme (MPSoC) sind in den letzten Jahren immer beliebter für industrielle Anwendungen aufgrund der hohen Performance, den niedrigeren Kosten und der Energieeffizienz geworden. Besonders die vielfältigen unterschiedlichen integrierten Prozessoren, auf denen unterschiedliche Betriebssysteme laufen, auch als Asymmetric Multiprocessing (AMP) bezeichnet, bringen viele Herausforderungen mit sich.

Die zwei größten Herausforderungen sind das Lifecycle-Management (LCM) und die Interprozessorkommunikation (IPC). Ein bekanntes Open Source Framework für diese Herausforderungen ist OpenAMP.

Diese Expert Session stellt den Aufbau von heterogenen MPSoCs und den Einsatz von unterschiedlichen Betriebssystemen auf MPSoCs vor. Anschließend wird OpenAMP vorgestellt und an einem konkreten Projektbeispiel umgesetzt. Das Projektbeispiel stellt die Portierung von dem OpenAMP Framework auf den eingesetzten MPSoC i.MX8 von NXP mit einem Embedded-Linux auf dem ARM Cortex-A35 und einem FreeRTOS auf dem ARM Cortex-M4 dar. Hierbei werden wichtige Randpunkte betrachtet und erläutert, die bei einer Portierung von großer Relevanz sind.

Zum Abschluss werden die Evaluierungsergebnisse zu auftretenden Latenzzeiten zwischen Linux-Userspace und FreeRTOS dargestellt und diskutiert ob, die umgesetzte AMP-Architektur für Echtzeit geeignet ist.

NUTZEN

Die Teilnehmer*innen erhalten einen Einblick in die Einsatzbereiche von heterogenen Multiprozessorsystemen (MPSoCs) und die dazu laufenden Betriebssysteme auf MPSoCs. Sie erhalten Anregungen, wie mögliche Architekturen auf MPSoCs aufgebaut sind, wie eine konkrete Implementierung anhand OpenAMP auf dem i.MX8 aufgebaut ist und die Darstellung vom Open Source Framework OpenAMP. Zusätzlich werden konkrete Evaluierungsergebnisse vorgestellt, damit Teilnehmer*innen die Echtzeitfähigkeit von OpenAMP auf dem i.MX8 einschätzen können.

ZIELGRUPPE

Entwicklung

Specification by Example

ABSTRACT

In diesem Vortrag geht es darum, wie ein Softwareentwicklungsteam gemeinsam mit allen Projektbeteiligten eine Spezifikation erstellen kann, die dazu führt, dass die tatsächlich gewünschte Softwarelösung in iterativen Schritten effizient geliefert wird. Dabei werden die benötigten Praktiken mit Hilfe eines fiktiven Entwicklungsprojektes, dem Spiel „Schiffe versenken“, praxisnah vorgestellt.

AGENDA

- Herausforderungen bei der Anforderungserhebung
- Herausforderungen bei der gemeinsamen Arbeit an Softwarelösungen
- Theoretische Übersicht über die Praktiken von „Specification by Example“
- Vorstellung der Herangehensweise anhand des Spiels „Schiffe versenken“
- Kollaboratives Design einer beispielorientierten Spezifikation mit Gherkin
- Erstellung einer „lebenden Spezifikation“ mit Hilfe von Testautomatisierung
- Vorteile und Herausforderungen einer „lebenden Dokumentation“

ZIELGRUPPE

Entwicklung, Test, Projektmanagement, Product Owner

Einführung in die Systemmodellierung mit SysML

ABSTRACT

Bei der Beschreibung komplexer Systeme haben Modelle zwar schon immer eine wichtige Rolle gespielt (zusammen mit dokumentbasiertem Arbeiten), doch erst die „Systems Modeling Language“ SysML bietet einen durchgängigen und ganzheitlichen, modellbasierten Ansatz für das Systems Engineering.

Wir zeigen anhand eines Beispiels, wie man mit der SysML Anforderungen und Spezifikationen aufnimmt, Strukturen und Verhalten eines Systems modelliert und mit Simulation, Design und Test verbindet. Dabei lernen wir alle relevanten Modellelemente und Diagrammtypen dieser Modellierungssprache kennen.

AGENDA

- Embedded Systems Engineering – was ist das?
- Modellbasiertes Systems Engineering
- Die SysML als Modellierungssprache – ein Überblick
- Anforderungen
- Strukturen
- Verhalten und Simulation

VORKENNTNISSE

Grundlegende Kenntnisse der Softwareentwicklung und der UML

ZIELGRUPPE

Entwicklung, Projektmanagement

Variantenmodellierung mit SysML

ABSTRACT

Wer immer ein Embedded System für mehr als einen Kunden entwickelt oder sein System besonders konfigurierbar machen will, muss Produktlinienengineering betreiben. Dabei kommt es leicht zu einer kombinatorischen Explosion von Requirements, Dokumentation, Modellen und Tests.

Wir zeigen anhand eines Beispiels, wie man mit der SysML als zugrundeliegender Notation für Systems Engineering gemeinsame Requirements und Features einer Produktfamilie modelliert und Varianten für besondere Anforderungen definiert. Dabei halten wir alles in einem Modell, sparen in der Entwicklung Zeit und Geld und erhöhen gleichzeitig die Qualität der Produktentwicklung.

AGENDA

- Embedded Systems Engineering
- Modellbasiertes Systems Engineering mit SysML
- Produktlinien – was ist das?
- Feature-Oriented Domain Analysis
- Variantenmodellierung, Vorteile und Probleme
- Orthogonal Variability Modeling

VORKENNTNISSE

Grundlegende Kenntnisse der UML, Interesse an Systems Engineering

ZIELGRUPPE

Softwareentwickler*innen, Systemingenieur*innen, Projektleiter*innen

SysML-Einführung für Systemingenieur*innen (Methodik, Sicherheitsbetrachtung, Funktionsarchitektur)

ABSTRACT

Wir zeigen, wie Systemingenieur*innen mit Modellen nicht nur die Anforderungen an ein System aufnehmen, sondern auch dessen Struktur und Verhalten beschreiben. Sie erhalten eine Einführung, wie Sie anhand von Modellen die Schnittstellen zu anderen Systemen darstellen, funktionale Sicherheit analysieren und dabei auch Faktoren wie Entwicklungs- und Betriebskosten, Wartung, Systemperformance, Test, Produktion und Wiederverwendung berücksichtigen können.

Dabei dient die Systems Modeling Language (SysML) als Grundlage für eine interdisziplinäre Systembeschreibung und eine ganzheitliche, modellbasierte Entwicklungsmethodik. Anforderungen, Analyse, Design, Simulationen und Tests werden mit Diagrammen und Modellelementen dieser Notation beschrieben und im Rahmen eines Beispiels erklärt.

AGENDA

- Überblick Prozess- und Projektziele
- Systeme, Embedded Systeme, Systems Engineering
- MBSE: Modellbasiertes Systems Engineering mit SysML
 - Funktionale Anforderungen
 - Systemkontext
 - Funktionales / logisches Modell
 - Konkretes Modell
 - Testen und Traceability
 - Nicht-funktionale Anforderungen

VORKENNTNISSE

Grundlegende Kenntnisse des Anforderungsmanagements und der UML

ZIELGRUPPE

Systemingenieur*innen, Softwareingenieur*innen, Qualitätsingenieur*innen, Projektmanagement

Effizienzsteigerung durch Scrum (Workshop)

ABSTRACT

Scrum ist ein Vorgehensmodell zur Bewältigung komplexer Entwicklungsprozesse. Die Arbeit soll während des Scrummens effektiver, Hindernisse dauerhaft beseitigt und die zeitliche Planung übersichtlicher werden. Der Vortrag erklärt, welche Vor- und Nachteile diese agile Methodik gegenüber der klassischen Projektplanung hat. Die an der Projektdurchführung beteiligten Personen nehmen verschiedene Rollen ein, um im Scrum-Prozess miteinander zu arbeiten. Auch die Zusammenarbeit mehrerer Teams in einem größeren Projekt wird erläutert. Bei komplexen Produktentwicklungen ist die Erstellung, Pflege und Umsetzung des Product Backlogs eine Herausforderung, die nicht zu unterschätzen ist.

Schilderungen aus der Praxis geben während des Vortrags einen Einblick, wie Scrum bei der Produktentwicklung eingesetzt werden kann und zeigen auch die Grenzen des Scrum-Prozesses auf. Fachbegriffe und Regeln des Scrum werden ausführlich erläutert, ebenso wie die benötigten Hilfsmittel zur Durchführung und Analyse.

AGENDA

- Agile Methodik
- Überblick über den Scrum-Prozess
- Grenzen des Scrum-Prozesses
- Aufgaben der am Scrum-Prozess beteiligten Personen
- Erstellung und Pflege des Product Backlogs
- Zusammenarbeit des Scrum-Teams
- Zusammenarbeit mehrerer Scrum-Teams
- Erklärung von Scrum-Artefakten und -Events
- Abschätzung der Zeitaufwände
- Kontrolle der abgeschätzten Zeitaufwände
- Aufbau und Benutzung eines Scrumboards
- Analyse des durchgeführten Sprints
- Best Practices
- Empfehlungen für die Einführung von Scrum
- Effizienzsteigerung der Arbeitsleistung
- Scrum of Scrums

ZIELGRUPPE

Projektmitarbeiter*innen und -leiter*innen, die mit der Entwicklung eines Produktes, der Durchführung eines Projektes oder der Erstellung einer Dienstleistung betraut sind.

Einführung in Embedded Systems (Workshop)

ABSTRACT

Von Glückwunschkarten mit Melodie über Kaffeeautomaten bis hin zu Marsrovern werden Steuerungen auf Basis von Embedded Systems eingesetzt. Diese Steuerungen werden in einer Kombination von hardwarenaher und abstrakter Software programmiert. Auf eine hardwarenahe Programmierung kann trotz moderner Programmiersprachen nicht verzichtet werden.

Diese Expert Session bietet einen Einblick in die Programmierung von Embedded Systemen. Es werden theoretische Grundlagen erörtert und in einer anschließenden praktischen Übung an einem Mikrocontroller verdeutlicht.

AGENDA

I Vortrag

- Einführung
- Mikrocontroller
- DSP (Digital Signalprozessor)
- FPGA, CPLD
- Programmierung
- Betriebssysteme
- Entwicklungstools
- Mikrocontroller-Aufbau
- Zusammenfassung

I Praktische Übung

- Installation der Entwicklungsumgebung
- Inbetriebnahme eines Mikrocontroller Evaluation-Boards
- Erste Schritte (z.B. Timer, GPIO´s, Schnittstellen)

VORKENNTNISSE

Programmierkenntnisse in C

TECHNISCHE VORAUSSETZUNGEN

Eigener Laptop (alle technischen Details erhalten Sie in Ihrer Teilnahmebestätigung)

ZIELGRUPPE

Entwicklung

Mikrocontroller (μ C) – Erste Schritte in der Programmierung (Workshop)

ABSTRACT

In unserem Workshop werden Sie mit der grundlegenden Funktionsweise und dem Aufbau von Mikrocontrollern vertraut gemacht. In einem anschließenden Hands-on-Training mit Programmierbeispielen wird der Umgang mit Mikrocontrollern und den zugehörigen Entwicklungsumgebungen erlernt.

AGENDA

- Einleitung
- Was ist ein Embedded System?
- Mikrocontroller-Grundlagen
- Hands-on-Training
 - Erstellen von Beispielprogrammen für diverse Peripherie-Units, wie z.B. Timer, I/O, UART
 - Debuggen des Source-Codes

VORKENNTNISSE

Grundlegende Kenntnisse der Softwareentwicklung und der Programmiersprache C

TECHNISCHE VORAUSSETZUNGEN

Eigener Laptop (alle technischen Details erhalten Sie in Ihrer Teilnahmebestätigung)

ZIELGRUPPE

Entwicklung

Field Programmable Gate Array (FPGA) für Einsteiger (Workshop)

ABSTRACT

Field Programmable Gate Arrays (FPGAs) werden seit vielen Jahren erfolgreich in der Mess-, Kommunikations- und Bildverarbeitungstechnik eingesetzt. Mit der stetigen Weiterentwicklung werden FPGAs auch im niedrigen Preissektor immer interessanter. Da sich der Designflow aber stark von dem für Mikrocontroller oder digitalen Signalprozessoren unterscheidet, zögern noch viele Entwickler*innen mit dem Einsatz von FPGAs. Dies möchten wir ändern.

Im Rahmen dieser Expert Session machen wir Sie mit den Grundlagen der digitalen Schaltungsentwicklung auf einem FPGA vertraut. Sie erhalten einen Überblick über die Funktionsweise und Einsatzmöglichkeiten von FPGAs und erstellen erste Designs mit der Intel Quartus Prime Toolchain, die Sie direkt auf einem Intel FPGA testen können.

AGENDA

I Vortrag

- Einführung in die digitale Schaltungsentwicklung
- Aufbau und Funktionsweise eines FPGAs
- Vorstellung des FPGA-Designflow
- Einführung in VHDL
- Einführung in die Synthese mit Intel Quartus Prime
- Simulation von VHDL-Modulen

I Praktische Übung

- Inbetriebnahme des Evaluation-Boards
- Erste Schritte (z.B. Counter)

VORKENNTNISSE

Grundlegende Kenntnisse der Softwareentwicklung und der digitalen Schaltungstechnik

TECHNISCHE VORAUSSETZUNGEN

Eigener Laptop (alle technischen Details erhalten Sie in Ihrer Teilnahmebestätigung)

ZIELGRUPPE

Entwicklung

Digitale Signalprozessoren (DSP) – Einführung in die Welt der DSPs (Workshop)

ABSTRACT

In unserem DSP-Workshop werden wir Sie zum einen mit den Grundbegriffen der analogen und digitalen Signalverarbeitung und zum anderen mit der Umsetzung von digitalen Filteralgorithmen auf einem ausgewählten DSP vertraut machen.

Neben einer Einführung in Scilabs, einem Werkzeug, das unter anderem zur Dimensionierung von digitalen Filtern am PC verwendet werden kann, werden Sie in diesem Workshop einen von Ihnen dimensionierten Filter auf einem DSP implementieren.

AGENDA

- Einführung in Signalverarbeitung und Scilab
- Erklärung unterschiedlicher (analoger) Filtertypen
- Möglichkeiten der digitalen Filter-Implementierung
- Einführung in Echtzeitbetriebssysteme
- Praxisbeispiel: Implementierung eines Filters auf realer Hardware nach vorhergehender Dimensionierung am PC

VORKENNTNISSE

Grundlegende Kenntnisse der Softwareentwicklung und der Programmiersprache C/C++

TECHNISCHE VORAUSSETZUNGEN

Eigener Laptop (alle technischen Details erhalten Sie in Ihrer Teilnahmebestätigung)

ZIELGRUPPE

Entwicklung

Echtzeitbetriebssysteme am Beispiel FreeRTOS

ABSTRACT

Projekte im Bereich der Embedded Software sind in den letzten Jahren stetig komplexer und umfangreicher geworden. Dies führt dazu, dass traditionelle Programmierungsmethoden immer mehr an ihre Grenzen kommen und man neue Wege finden muss, mitunter zeitkritische Applikationen zu erstellen. An dieser Stelle wird oft der Ruf nach Echtzeitbetriebssystemen laut, doch was bieten uns diese wirklich? Diese Frage soll in unserer 1,5-stündigen Expert Session am Beispiel „FreeRTOS“ beantwortet werden.

AGENDA

- Einführung
 - Was ist Echtzeit?
 - Unterscheidung zwischen harter, weicher und fester Echtzeit
 - Vergleich unterschiedlicher Embedded Software-Programmierungsansätze
- Was ist FreeRTOS?
- Was bietet FreeRTOS?
 - Tasks und Co-Routines
 - Scheduling
 - Queues / Mailboxes
 - Semaphore und Mutexes
 - Software Timer
 - Advanced Features
- Echtzeitbetriebssystem verwenden: ja oder nein?
- Zusammenfassung

VORKENNTNISSE

Grundlegende Kenntnisse der Softwareentwicklung

ZIELGRUPPE

Technisches Projektmanagement, Entwicklung

Grundlagen der Echtzeitsysteme

ABSTRACT

Echtzeitsysteme finden breite Anwendung in den Gebieten der Mess-, Steuer- und Regelungstechnik und in Multimediasystemen. Für die dort vorhandenen zeitkritischen Prozesse und Aufgaben müssen die einzusetzenden Betriebsmittel und Protokolle Bedingungen erfüllen, die sich nicht nur auf die Laufzeitumgebung, sondern auch bereits auf die Planung und Entwicklung auswirken.

Dieser Vortrag erläutert einige der Anforderungen an Echtzeitsysteme und stellt die UML und die Berechnung von Echtzeitverhalten von solchen Systemen vor.

AGENDA

- Einführung: grundlegende Probleme bei Echtzeitsystemen
- Vorstellung des UML MARTE Profils
- Konzepte zur Berechnung und Nachweis von Echtzeitverhalten
- Vom Modell zum Code: Programmieren mit POSIX Echtzeiterweiterungen
- Verbinden von Systemen: RT-CORBA und Echtzeitnetzwerke

VORKENNTNISSE

Grundlegende Kenntnisse von Betriebssystemen, Netzwerken und UML

ZIELGRUPPE

Entwicklung

Erfolgreiche Softwareprojekte durch Requirements Engineering

ABSTRACT

Es gibt immer wieder Softwareprojekte, die fehlschlagen – sei es, dass sie nicht zu Ende geführt oder die Projektziele verfehlt werden. Dieser Vortrag beschäftigt sich mit der Entstehung von Fehlern und zeigt auf, wie man diese Fehlerquellen durch Requirements Engineering vermeiden kann.

Der dritte Teil der Präsentation dreht sich um Use Cases und beschäftigt sich mit der Frage „Was sind Use Cases und wie verläuft der Weg von der Use Cases-Erstellung bis hin zur Ableitung der Requirements?“.

AGENDA

- Erfolg oder Misserfolg im Softwareprojekt
 - Ziele im Softwareprojekt
 - Fehler & Kosten
 - Gründe für gescheiterte Softwareprojekte
- Requirements Engineering als Antwort
 - Arten von Requirements
 - Fehlervermeidung durch Requirements
 - Die ersten Schritte beim Requirements Engineering
- Use Cases im Requirements Engineering
 - Was ist ein Use Case?
 - Use Cases erstellen
 - Requirements aus einem Use Case extrahieren

ZIELGRUPPE

Entwicklung, Projektmanagement

Requirements Analyse mit Use Cases

Teil 1: Einführung in Requirements Engineering

ABSTRACT

Anforderungen an Systeme sind in der Analysephase oft noch nicht klar ersichtlich oder sind aus den vorliegenden Informationen schwer zu extrahieren. Eine systematische Vorgehensweise und die richtige Methodik spielen eine entscheidende Rolle, um User- und System-Requirements zu finden und zu formulieren.

Der erste Teil der Vortragsreihe bietet eine allgemeine Einführung in das Themengebiet Requirements Engineering und Analyse. Was sind die Requirements, wo und wann treten sie in verschiedenen Entwicklungsprozessen auf? Welche Typen von Requirements werden unterschieden und welche Möglichkeiten und Strategien eignen sich zu deren Erfassung?

AGENDA

- Einführung: Requirements
- Entwicklungsprozesse
 - Phasen des Entwicklungsprozesses
 - Prozessmodelle
 - Requirements im jeweiligen Prozess
 - Change Management
- Requirements
 - Funktionale und nicht-funktionale Requirements
 - Constraints
 - FURPS+
 - Lebenszyklen von Requirements
 - Requirements und Use Cases
- Requirement-Analyse
 - Stakeholder
 - Akteure
 - Systemkontext
 - Benutzerziele
 - Use Cases
 - Ergänzende Spezifikation
- Zusammenfassung

VORKENNTNISSE

Grundlegende Kenntnisse der Softwareentwicklung

ZIELGRUPPE

Entwicklung, Projektmanagement

Requirements Analyse mit Use Cases

Teil 2: Requirements mit Use Cases

ABSTRACT

Anforderungen an Systeme sind in der Analysephase oft noch nicht klar ersichtlich oder aus den vorliegenden Informationen schwer zu extrahieren. Eine systematische Vorgehensweise und die richtige Methodik spielen eine entscheidende Rolle, um User- und System-Requirements zu finden und zu formulieren.

Der zweite Teil der Vortragsreihe beschäftigt sich detailliert mit dem Konzept der Use Cases als zentrales Element zur Aufdeckung von Requirements. Was sind Use Cases und welche Parallelen gibt es zur UML? Welche Typen von Use Cases werden unterschieden und wie können diese effektiv erstellt werden? Ein abschließendes Resümee soll zeigen, warum Use Cases der gegenwärtige Standard zur Erfassung von Requirements sind.

AGENDA

- Einführung: Use Cases
- Use Cases in der Unified Modeling Language (UML)
- Use Case Typen
 - Betrachtungsfokus, Geltungsbereich, Sichtweite
 - Formalitäten, Ebenen
- Use Cases erstellen
 - Use Case Templates
 - Szenarien
 - Exceptions
 - Formulierung
 - Technische Details in Use Cases
- Use Cases und System Requirements
- Vor- und Nachteile von Use Cases

VORKENNTNISSE

Grundlegende Kenntnisse der Softwareentwicklung. Die Teilnahme an Teil 1 (Einführung in das Requirements Engineering) der Vortragsreihe wird empfohlen.

ZIELGRUPPE

Entwicklung, Projektmanagement

Requirements Analyse mit Use Cases

Teil 3: Workshop Requirements Analyse

ABSTRACT

Anforderungen an Systeme sind in der Analysephase oft noch nicht klar ersichtlich oder aus den vorliegenden Informationen schwer zu extrahieren. Eine systematische Vorgehensweise und die richtige Methodik spielen eine entscheidende Rolle, um User- und System-Requirements zu finden und zu formulieren.

Im dritten Teil werden in einer konkreten Aufgabenstellung die in den beiden ersten Teilen der Vortragsreihe vorgestellten Konzepte durch die Teilnehmer*innen umgesetzt. Dabei werden in Anlehnung an einen iterativen und inkrementellen Entwicklungsprozess in einem ersten Durchgang Artefakte angelegt, die in einem zweiten Durchgang erweitert und verfeinert werden.

AGENDA

- Artefakte im Entwicklungszyklus
- Aufgabenstellung
- Lösungsweg in Gruppenarbeit (1. Iteration)
 - Vision
 - Stakeholder, Akteure
 - User Requirements formulieren
 - Use Case Diagramm
 - Funktionale System Requirements mit Use Cases
 - Nicht-funktionale Requirements
- Lösungsweg in Gruppenarbeit (2. Iteration)
 - Erweiterung und Verfeinerung der Artefakte aus der 1. Iteration
 - Berücksichtigung neu gefundener Requirements

VORKENNTNISSE

Grundlegende Kenntnisse der Softwareentwicklung. Die Teilnahme an den ersten beiden Teilen (Einführung in das Requirements Engineering und Requirements mit Use Cases) der Vortragsreihe wird empfohlen.

ZIELGRUPPE

Entwicklung, Projektmanagement

Brauchen agile Teams noch Führungskräfte?

ABSTRACT

Der Weg von der klassischen zur agilen Organisation bringt nicht nur Strukturänderungen mit sich, auch Rollen und Verantwortungen werden beeinflusst.

Neue Rollen werden eingeführt, vorhandene abgeschafft und statt klaren Anweisungen werden selbstorganisierende Teams gebildet.

Braucht es dann überhaupt noch Führungskräfte im klassischen Sinne?

Holger Neumann zeigt mit diesem Vortrag, warum Führungskräfte trotz der veränderten Strukturen essentiell sind und wie sich durch die Transformation zwangsläufig auch die Rolle des Managements verändert.

ZIELGRUPPE

Projektmanagement, Führungskräfte

Linux-Grundlagen (Workshop)

ABSTRACT

Diesen Grundlagenworkshop bieten wir in einer kostenlosen halbtägigen Version an und in einer ausführlichen zweitägigen Version.

Wie ist ein Linux-System aufgebaut? Wie sieht das Filesystem aus? Wie erledige ich grundlegende Arbeiten auf dem System? Nach diesem zweitägigen Workshop haben Sie die Fähigkeiten sich effizient auf einem Linux-System zu bewegen. Außerdem erlernen Sie die Grundlagen der Shell-Programmierung, IP-Konfiguration und Paket-Administration. Anhand von praktischen Übungen auf einem VirtualBox Image werden die erlernten Kenntnisse praktisch angewendet.

AGENDA

- Historie und Architektur
 - Wie ist Linux entstanden?
 - Die System-Architektur von Linux
- Die Shell – Schnittstelle zum System
 - Was ist die Shell und was bietet sie mir?
 - Einführung in die wichtigsten Shell-Kommandos wie find, grep, tar und zip
 - Grundlagen für die Bedienung des Texteditors Vi
 - Grundlagen der automatisierten Verarbeitung von Texten (tr, cut, grep, sort)
- Der Linux-Dateibaum
 - Wie ist das Linux-Dateisystem aufgebaut?
 - Einführung in die Filesystem-Befehle der Shell
- Die Umgebungsvariablen
 - Was sind Umgebungsvariablen und wie nutze ich sie?
- Fortgeschrittene Bash Features
 - IO-Redirection und Pipelining
- Das Linux-Rechtmanagement
 - Einführung in den Aufbau des Linux-Rechtesystems
 - Vergabe und Kontrolle von User-, Gruppen- und Dateirechten

TECHNISCHE VORAUSSETZUNGEN

Eigener Laptop (alle technischen Details erhalten Sie in Ihrer Teilnahmebestätigung)

ZIELGRUPPE

Entwicklung

Debug your Linux – Praktisches Debuggen auf echter Hardware (Workshop)

Teil 1: GDB

ABSTRACT

Am Ende der Entwicklung steht die Fehlersuche. Haben Sie die Tools im Griff? Wir helfen Ihnen, den GDB als den Standard-Debugger unter Linux zu beherrschen. Egal, ob von der Kommandozeile oder aus Eclipse heraus. Die meisten Übungen werden auf dem Raspberry Pi, das im Workshop-Preis enthalten ist, durchgeführt. So können Sie die Übungen zuhause jederzeit noch einmal nachvollziehen.

AGENDA

- Einführung in das Raspberry Pi
 - Features des Boards
 - Kommen Sie in Kontakt mit Ihrem Board
- GDB: der GNU Debugger auf der Kommandozeile
 - Einzige Voraussetzung: eine Shell
 - Remote Debugging auf dem Target
 - Einfaches Setup
- GUI Frontend zu GDB: mehr Komfort, mehr Übersicht
 - cgdb, Konsolen-GUI für den GDB
- Grundfunktionen des GDB
 - Debug-Sitzungen
 - Breakpoints, Watchpoints, Catchpoints
 - Stack Analyse
- Cross-GDB: wenn auf dem Target kein Platz für einen Debugger ist
 - Kommandozeilen-GDB als Cross Debugger
 - Cross Debugging auf dem Raspberry Pi
- Thread Debugging
 - Multithreaded Programme debuggen
 - Fehlersuche in einem Thread, während andere weiterlaufen
- Debugging mit Eclipse auf dem Raspberry Pi
 - Effektives Remote- und Crossdebugging mit Eclipse
 - Schritt für Schritt setzen wir ein Debug-Projekt auf
- Feature Staccato
 - Postmortem Debugging
 - Reverse Debugging

VORKENNTNISSE

Grundlegende Kenntnisse der Softwareentwicklung und der Nutzung von Linux in der Shell sowie sicherer Umgang mit C/C++ sind Voraussetzung. Erfahrungen in der Entwicklung von Embedded Systems sowie Kenntnisse der Linux-Kernelprogrammierung sind von Vorteil.

TECHNISCHE VORAUSSETZUNGEN

Eigener Laptop (alle technischen Details erhalten Sie in Ihrer Teilnahmebestätigung)

ZIELGRUPPE

Entwicklung

Debug your Linux – Praktisches Debuggen auf echter Hardware (Workshop)

Teil 2: Systemtap, ftrace und Valgrind

ABSTRACT

Besiegen Sie Ihre Speicherfehler mit Valgrind. Kommt Embedded Hardware ins Spiel, ist ein systemweiter Ansatz zur Fehlersuche gefragt. Die Fehlersuche bei Embedded Applikationen macht vor dem Kernel nicht halt. Mit Systemtap und ftrace lernen Sie zwei einfache, aber leistungsfähige Tools kennen.

AGENDA

- Valgrind: Aufspüren von Speicherproblemen (nur auf x86)
 - Fehler bei malloc / free (auch double free)
 - Uninitialisierte Variablen
 - Memory Leaks
- Cachegrind: Cache Profiling (nur auf x86)
 - Detaillierte Simulation der L1, D1 und L2 Caches
 - Statistiken zu Cache Misses mit Source Zeilen / Funktionen
- Callgrind: Callgraph Profiling (nur auf x86)
 - Analyse der Aufrufreihenfolgen
 - KCachegrind: Grafische Darstellung der Performance-Analyse
- Systemtap-Framework
 - Kprobes, was ist das?
 - Einführung in das Systemtap-Framework
- Systemtap Script-Sprache
 - Tracing im Kernel mit einer Script-Sprache
 - Kernelprogrammierung fast ohne Kernel-Know-how
- Schreiben Sie Ihr eigenes Systemtap-Script
 - Tracing von Kernelmodul-Funktionen
 - Tracing von Userspace Applikationen (nur auf x86)
- ftrace
 - Läuft auf jeder Linuxplattform
 - Steuerung von ftrace auf der Kommandozeile
- Spezielle Tracer
 - Was macht mein Scheduler und warum?
 - Wo bleiben meine Interrupts?
- Weitere ftrace-Features
 - Die Suche nach der Stecknadel im Heuhaufen
 - Filtern von Trace-Ereignissen
 - Aktivieren / Deaktivieren des Tracing aus dem Userspace
 - trace-cmd
 - Der Kernelshark

VORKENNTNISSE

Grundlegende Kenntnisse der Softwareentwicklung und der Nutzung von Linux in der Shell sowie sicherer Umgang mit C/C++ sind Voraussetzung. Erfahrungen in der Entwicklung von Embedded Systems sowie Kenntnisse der Linux Kernelprogrammierung sind von Vorteil. Der Besuch des 1. Teils des Workshops (GDB) wird empfohlen.

TECHNISCHE VORAUSSETZUNGEN

Eigener Laptop (alle technischen Details erhalten Sie in Ihrer Teilnahmebestätigung)

ZIELGRUPPE

Entwicklung

Application Programming Kick-Start

ABSTRACT

Der Einstieg in die Software-Entwicklung unter Linux fällt Ihnen schwer? Wir gehen mit Ihnen gemeinsam die ersten Schritte am Beispiel des Raspberry Pi und dem Zusatzboard Pibrella.

In der Expert Session zeigen wir Ihnen das Arbeiten mit der POSIX API und den Umgang mit dem Debugging-Werkzeug GDB. Außerdem lernen Sie die wichtigsten Methoden wie Remote & Cross Debugging und die Hardware-Ansteuerung unter Linux anhand einfacher Beispiele kennen.

AGENDA

- Using command-line version of GDB
- Using command-line of GDB
- Debug sessions
- Breakpoints and catchpoints
- Stack analysis
- Remote Debugging and Cross Debugging
- Remote Debugging with Eclipse
- Einstieg in die Linux-Programmierung am Beispiel der Nutzung von GPIOs und poll/select
- Hardware Raspberry Pi mit Pibrella Board

VORKENNTNISSE

Erfahrungen im Umgang mit Linux auf der Shell/Desktop, Programmierkenntnisse in C/C++

ZIELGRUPPE

Entwickler*innen, die Erfahrung in der Applikationsentwicklung haben und den ersten Einstieg in die Applikationsentwicklung unter Linux bewältigen wollen.

Entwicklungsbegleitend Testen mit dem Mixed Mode Hardware-in-the-Loop Framework (HiLF)

ABSTRACT

Jede Entwicklungsabteilung kennt das Problem: Software muss auf die Zielhardware integriert und getestet werden. Häufig ist aber die Hardware nicht verfügbar (noch nicht fertig, begehrter Prototyp, etc.) oder allen Entwickler*innen zugänglich. Die Emulation nicht vorhandener Hardware schafft hier Abhilfe.

Das HiL-Framework wurde konzipiert, um ein äußerst flexibles und kostengünstiges Testframework mit Hardware-Emulation/Simulation bereitzustellen (keine Lizenzgebühren!). Es enthält diverse Bausteine, die je nach Bedürfnissen und Anforderungen zusammengestellt werden. Schnittstellen wie CAN, UART, SPI und viele weitere sind als Module vorhanden und können verwendet werden, um Systeme funktional zu testen.

AGENDA

- Was ist Hardware-in-the-Loop?
- Was ist das entwicklungsbegleitende HiLF von Mixed Mode?
- CI und Cloud-Einsatzmöglichkeiten mit dem HiLF
- Demonstration des HiLF auf einer Embedded-Plattform

ZIELGRUPPE

Entwicklung, Test, Projektmanagement, Product Owner

Security for Safety am Beispiel der IEC 62443 im Kontext zur IEC 61508

ABSTRACT

Im Zeitalter der Digitalisierung (Industrie 4.0) stellt sich die Frage, wie man das Thema Cyber-Security bei einer Safety-Entwicklung mit berücksichtigen sollte. Die Norm IEC 62443 bietet für das industrielle Umfeld einen Ansatz, Risiken zu bewerten und für Produkte, entsprechend des definierten Risiko-Levels, geeignete Gegenmaßnahmen zu treffen, um das Risiko auf ein akzeptables Maß zu begrenzen. Besonders interessant ist es, wenn das Produkt aus dem Safety-Umfeld kommt.

Der Vortrag soll Aufschluss geben, was man zu beachten hat, wenn man Security und Safety kombiniert.

AGENDA

- Gibt es im Industrie-Umfeld wirklich Cyber-Angriffe? (Beispiele...)
- Welche Tools verwenden Hacker und wie gehen diese vor?
- Wie kann man Cyber-Security-Risiko bestimmen und wie steht es in Relation zum Safety-Risiko?
- Vorstellung der IEC 62443 Normen-Reihe mit Fokus auf die Produkt-Entwicklung
- Vorgaben an einen Entwicklungsprozess, um sichere Produkte zu entwickeln. Wie kann ich meinen bestehenden Safety-Entwicklungsprozess sinnvoll ergänzen?
- Überblick über die geforderten Techniken und Maßnahmen, um einer Cyber-Angriff zu erschweren
- Diskussion: Wie kann ich in meiner Firma mit dem Thema Cyber-Security starten. Macht ein Produkt-Assessment Sinn? Wie kann ich den Safety Life-Cycle sinnvoll ergänzen, um auch das Thema Cyber-Security abzudecken?

Einstieg in die TrustZone mit OP-TEE

ABSTRACT

Nahezu alle modernen Prozessorarchitekturen stellen einen besonderen Betriebsmodus für die sichere Ausführung von Software bereit. Im Embedded- und IoT-Bereich ist die Arm TrustZone-Technologie dabei die bekannteste und dominierende Variante eines solchen Trusted Execution Environments (TEE).

Für die Verwendung dieses speziellen Betriebsmodus ist jedoch in der Regel viel hardwarenaher Code vonnöten. Während große Anbieter von z.B. Mobiltelefonen noch eigene Frameworks und Werkzeuge entwickeln können, steht solch eine Herangehensweise bei vielen Embedded-Projekten nicht zur Debatte.

Auf der anderen Seite soll Software wiederverwendbar und gegen eine möglichst stabile API entwickelt werden. Das Global Platform-Konsortium pflegt hierfür die Spezifikation einer TEE-API. Das fehlende Bindeglied und Kern des Beitrags ist die Implementierung dieser Spezifikation auf der TrustZone-Plattform.

Das Open Portable Trusted Execution Environment (OP-TEE) ist eine solche Implementierung. Sie unterstützt zahlreiche Plattformen vom Emulator über den Raspberry Pi bis zum Xilinx UltraScale+ Zynq MPSoC. Dabei sind gegen die TEE-API geschriebene Trusted Apps auf allen unterstützten Plattformen lauffähig. Das erhöht die Wiederverwendbarkeit von geschriebenem Code. Vor allem ermöglicht es aber eine schnelle, unkomplizierte und unabhängige Entwicklung von TrustZone-Anwendungen für eingebettete Systeme.

Unsere Expert Session stellt Konzept und Verwendung von OP-TEE vor und zeigt an einem kleinen Beispiel, wie schnell sich sichere Anwendungen damit umsetzen lassen. Die Zuhörer*innen erhalten einen kompakten Überblick über die Entwicklung von TrustZone-Anwendungen mit OP-TEE und sind anschließend in der Lage, selbst mit OP-TEE-Anwendungen zu experimentieren und davon ausgehend komplexere Anwendungen zu entwickeln.

ZIELGRUPPE

Entwicklung, Projektmanagement

Advanced Security mit Hardware Security Modulen

ABSTRACT

Je mehr die Welt vernetzt ist, umso mehr steigt auch die Gefahr durch Cyberangriffe. Um Anwendungen gegen Cyberangriffe zu schützen, müssen die Geräte und Kommunikationswege geschützt werden. Hier kommen meist in Software umgesetzte Kryptographie-Algorithmen zum Einsatz. Doch schützen diese nicht immer vor Angriffen direkt am Gerät. Es gibt aber auch hier Wege, z. B. durch eine Turn Key Solution diese Informationen zu schützen.

Im Rahmen dieser Expert Session erfahren Sie, wie Kommunikationswege mittels Hardware Security Modulen gesichert werden können.

AGENDA

- Anwendungsgebiete und Herausforderungen
- Grundlagen der sicheren Kommunikation
- Hardware Security Module
- Demonstration

ZIELGRUPPE

Entwicklung, Projektmanagement

Embedded Security mit FPGAs - Dynamische partielle Rekonfiguration am Beispiel von Xilinx-FPGAs

ABSTRACT

Implementierungen von Algorithmen auf FPGAs bieten vielfältige Vorteile: Hohe Performance bei gleichzeitig hoher Flexibilität und Erweiterbarkeit.

Weiterhin benötigen FPGA-Implementierungen spezielle Schutzmechanismen gegen Angriffe aufgrund der im Vergleich zu einem ASIC größeren Strukturen. Gerade kryptographische Algorithmen sind besonders anfällig für Seitenkanalangriffe. Andere Algorithmen kapseln oft essentielle IP des Unternehmens, die fremden Augen verborgen bleiben soll.

Hier kann ein durch Seitenkanalangriffe gestütztes Reverse-Engineering zu einem großes Problem werden. Dynamische partielle Rekonfiguration kann in beiden Fällen den Aufwand für den Angreifer deutlich erhöhen.

Hierfür werden während der Synthese des Designs für den zu schützenden Teil mehrere funktional äquivalente Logikvarianten generiert und in partiellen Bitfiles abgelegt. Diese können zur Laufzeit zufällig von einem (Mikro-)Controller ausgewählt und platziert werden.

Da ein Angreifer nicht auf triviale Weise unterscheiden kann, welche Variante zum Zeitpunkt der Seitenkanalmessung aktiv ist, erhöhen sich die Anzahl der benötigten Messungen und der Aufwand des Postprocessings für einen erfolgreichen Angriff signifikant. Zusätzlich kann auch der Ort der Platzierung variiert werden, um die Anzahl der Möglichkeiten weiter zu erhöhen. Insbesondere in Kombination mit herkömmlichen Maßnahmen ist ein deutlich verbesserter Schutz gegen aktive und passive Seitenkanalangriffe bzw. Reverse Engineering zu erwarten.

Dazu entwickelt Mixed Mode im Forschungsprojekt SecRec einen toolgestützten Workflow, der es ermöglicht, automatisiert partielle Bitfiles zu erzeugen. Diese Bitfiles werden über die TrustZone verschlüsselt in den FPGA geladen. Zudem wird dargelegt, wie die partielle Rekonfiguration auch außerhalb des Security-Bereichs, wie z.B. in der Bildverarbeitung verwendet werden kann.

ZIELGRUPPE

Entwicklung, Systemingenieur*innen

Vom ITSM zum Sicheren Eingebetteten System

ABSTRACT

Sichere Eingebettete Systeme zu entwickeln ist alles andere als leicht.

Unsere Expert Session gibt einen Überblick über die Voraussetzungen, Werkzeuge und Methoden zur Entwicklung Sicherer Eingebetteter Systeme. Die erste Erkenntnis ist, dass man keine sichere Software ohne sichere Entwicklungsumgebung entwickeln kann. Das ist die Motivation für die Einführung eines systematischen IT-Sicherheits-Managements. Die zweite Erkenntnis ist, dass man nicht einfach „drauflos programmieren“ kann. Zuvor steht ein Systemdesign, das Security und Privacy berücksichtigt. Schon während dieses Design entsteht, kann und soll man es mit (semi) formalen Methoden prüfen. STRIDE und DREAD zählen hierfür zu den prominentesten Ansätzen. Beide werden in der Expert Session eingeführt.

Auf der anderen Seite endet der Entwicklungsprozess nicht, sobald der Code kompiliert. Am Beispiel des Microsoft Security Development Lifecycle besprechen wir zusätzliche Maßnahmen für gute und sicherere Software.

Zum Abschluss wenden wir uns einem echten Stück Software zu. Wir modellieren die vorgefundene Situation im MS Thread Modellig Tool, identifizieren Bedrohungen und Schwachstellen, demonstrieren die zugehörigen Angriffe am Beispiel und setzen die notwendigen Gegenmaßnahmen in Code um.

ZIELGRUPPE

Entwicklung, Projektmanagement

Sicherheitslösungen für ressourcenbeschränkte Mikrocontroller

ABSTRACT

IoT-Geräte übertragen häufig sensible Daten und müssen abgesichert werden. Dies kann durch eine starke Verschlüsselung in Kombination mit sicherer Hardware erreicht werden.

Da viele „Dinge“ im IoT allerdings nur wenig rechenintensive Arbeiten verrichten und zudem mit einem möglichst geringen Energieverbrauch auskommen sollen, kommen häufig leistungsschwache Kleinstcontroller zum Einsatz. Diese sind zwar für ihre Aufgabe spezialisiert und ausreichend, für komplexe Sicherheitsfunktionen jedoch in der Regel zu schwach.

In dieser Expert Session erfahren Sie, wie auf leistungsschwachen Systemen trotz ihrer Einschränkungen verschiedene Security-Anforderungen konzeptioniert und umgesetzt werden können.

AGENDA

- Anwendungsgebiete und Herausforderungen
- Vergleich von Kommunikationsverfahren
 - IPSec
 - MACSec
 - TLS/SSL
- Vergleich geeigneter Software-Bibliotheken
 - Freie und kommerzielle Lösungen
 - Embedded Linux und Baremetal
 - OpenSSL, mbedTLS, MatrixSSL, NanoSSL, escript,...
- Performance von Kryptographie auf Mikrocontrollern
 - Softwareimplementierung
 - Hardwarebeschleunigung

ZIELGRUPPE

Entwicklung, Projektmanagement

Agile and Safety – Can this work?

ABSTRACT

„Safety first“ war und ist das Mantra in der Automobil-Branche mit dem Ziel, Menschenleben zu schützen. Die stetig zunehmende Vernetzung und Komplexität der Autos zwingt auch die Automobil-Industrie über alternative Entwicklungs-Methoden zur „klassischen Wasserfall-Entwicklung“ nachzudenken. Agile, Scrum & Co wird nun auch immer öfters im Functional Safety-Bereich eingesetzt, aber ist das nicht ein Widerspruch?

In dem 2-stündigen Vortrag werden wir gemeinsam die Widersprüche zwischen den Anforderungen des Functional Safety Standards (ISO26262 bzw. ASPICE) und den Werten und Prinzipien von Agile erarbeiten. Anschließend werden wir auf Basis des SCRUM-Frameworks diskutieren, welche Erweiterungen bzw. Änderungen notwendig wären, um den Safety-Ansprüchen gerecht zu werden und um einem Assessment standhalten zu können.

VORKENNTNISSE

Grundkenntnisse über SCRUM/ agile Entwicklungsprozesse

ZIELGRUPPE

Entwicklung, Projektmanagement

GUI-Gestaltung

ABSTRACT

Was ist ein GUI und was bringt eine ergonomische GUI-Gestaltung?

Graphical User Interfaces (GUI) sind im Idealfall eine hocheffiziente Schnittstelle zwischen Anwender*in und System. Gelungene GUIs zeichnen sich durch eine intuitive, fließende Handhabung aus, die die Anwender*innen in ihrer Arbeit unterstützt. Im Gegenzug erfordert ergonomisches GUI-Design die besondere Berücksichtigung der Persönlichkeit und der Arbeitsweise der Anwender*innen.

Dieser Vortrag erläutert die GUI-Gestaltung in der Praxis. Ausgehend von Begriffsdefinitionen wird der Nutzen ergonomischen GUI-Designs anhand von Beispielen aufgezeigt.

AGENDA

- Was sind GUIs?
 - Begriffsklärung
 - Beispiele
- Ergonomie
- Benutzbarkeit versus Gebrauchstauglichkeit
 - Was bedeutet Gebrauchstauglichkeit?
 - Erläuterung und Abgrenzung der Begriffe
- Produktqualität & Nutzungsqualität
 - Prozess- & Produktsicht
 - QM klassisch, QM ganzheitlich
- Grundsätze der Dialoggestaltung
 - Wie werden Arbeitsziele effektiv, effizient und zufriedenstellend erreicht?
 - Beispiele
- Ergonomische Maskengestaltung
 - Visuelle Platzierung und Ausrichtung von Elementen
 - Optimale Gruppierung und Anordnung von Elementen
 - Schriftwahl, Lesbarkeit und Verwechslungsgefahr
 - Farbgestaltung
- Gesteninteraktion
- Beurteilung der Softwareergonomie

VORKENNTNISSE

Grundlegende Kenntnisse der Softwareentwicklung

ZIELGRUPPE

Entwicklung, Projektmanagement

Plattformübergreifende App-Entwicklung

ABSTRACT

In unserer Expert Session bieten wir einen Überblick über verschiedene Ansätze zur Entwicklung von mobilen Apps. Native, hybride oder webbasierte App-Entwicklung? Am Beispiel der App des Gasprozessmanagementsystems „*e-Gas*“ der PIXEL GmbH werden die verschiedenen Ansätze vorgestellt und miteinander verglichen.

Des Weiteren werden u.a. Themen wie App-Design, -Distribution und technische Infrastruktur behandelt.

AGENDA

- Überblick mobile Plattformen
- Was ist Cross-Plattform-Entwicklung?
- Vergleich verschiedener Ansätze:
 - Nativ
 - HTML5 App
 - Hybrid
 - Laufzeitumgebung
 - Terminal App
 - Cross Compiler
 - Responsive UI
- Demonstration der *e-Gas*-App
- Verteilung von Apps
- Voraussetzungen für die Infrastruktur
- Sicherheitsaspekte
- Entwicklungstrend

VORKENNTNISSE

Interesse an mobilen Applikationen

ZIELGRUPPE

Entwicklung, Entscheider*innen mit technischem Hintergrund

Softwarecode-Metriken

ABSTRACT

Im ersten Teil des Vortrags wird das Thema „statische Codeanalyse“ behandelt. Die gezeigten Methoden sind „sprachenunabhängig“ und werden am Beispiel von C erläutert. Bei dieser Expert Session lernen Sie bekannte und praxiserprobte Software-Metriken kennen. Zur Ermittlung der Metriken benötigt man lediglich den Quelltext eines Programmes und die vorgestellten Tools.

Der zweite Teil des Vortrags dreht sich um die dynamische Codeanalyse und beantwortet unter anderem, welche Aussagekraft Software-Metriken besitzen.

AGENDA

I Statische Codeanalyse

- Wie „groß“ ist eine Software?
- Wie kann man die Komplexität von Software messen?
- Wie kann man zeigen, warum manche Aufgaben mehr Zeit brauchen?

I Dynamische Codeanalyse

- Wie kann man messen, wie gut getestet wurde?
- Wie findet man „Lücken im Test“?
- Wie kann man die Anzahl der benötigten Unit Tests für Funktionen / Module schätzen?
- Welche Aussagekraft besitzen die entsprechenden Software-Metriken?

VORKENNTNISSE

Kenntnisse in C

ZIELGRUPPE

Entwicklung

Code-Metriken konkret: Softwarequalität erfassen

In der 1-stündigen kostenfreien Expert Session werden die wichtigsten Software-Metriken erklärt. Dabei geht es um folgende Fragen aus der täglichen Praxis von Software-Entwickler*innen/-Tester*innen:

- Welche Aussagekraft besitzen Software-Metriken?
- Wie kann man die Komplexität von Software messen?
- Wie findet man „Lücken im Test“?
- Wie kann man messen, wie gut getestet wurde?
- Wie kann man den Testaufwand mit Hilfe von Software-Metriken schätzen?

Legacy Code: Wie teste ich chaotische Software? (Workshop)

Teil 1: Grundlagen, Problematik, Grundbegriffe der Software-Metriken

ABSTRACT

Das zweitägige Seminar befasst sich mit der Vermessung sowie dem dynamischen Test von Legacy Code, wobei hier der Begriff „Legacy Code“ enger gefasst wird auf alten, überlieferten Programmcode, der erkennbar gewachsen ist anstatt geplant erstellt worden zu sein, und zu dem Kommentare und Spezifikationen weitgehend fehlen.

Motivation für dieses Seminar ist die vielfach in der Praxis gemachte Beobachtung, dass entgegen den in Schulungen und Literatur vertretenen Ansätzen, die Codierung vielfach nicht von wohldefinierten und reviewten Spezifikationen ausgeht und nicht entlang geordneter Prozesse und Arbeitsabläufe geschieht. Vielmehr ist in höherem Maße, als es den Anschein hat, Codierung auf Zuruf unter hohem Zeitdruck bei knapper Personaldecke und unzureichendem Wissen die Realität in kleineren und größeren Betrieben. Solche Vorgehensweisen resultieren in dem, was wir als „logisches Gestrüpp“ bezeichnen.

Daraus ergibt sich die Notwendigkeit, Code dieser Beschaffenheit vermessen und testen zu müssen, um die verschiedenen logischen Aspekte des Codes verstehen und eine objektive Qualitätsaussage machen zu können. Aus verschiedenen Gründen können die eigentlich notwendigen Redesigns nicht durchgeführt werden. Deshalb müssen andere Wege gefunden werden, um zu einer belastbaren Qualitätsaussage zu kommen.

Dieses Seminar ist darauf ausgerichtet, die Teilnehmer*innen für die Thematik des logischen Gestrüpps zu sensibilisieren und an die Problematik seiner eingeschränkten Handhabbarkeit hinzuführen. In theoretischen und größtenteils praktischen Übungen sollen die Teilnehmer*innen mit Methodiken bekanntgemacht werden, um Code dieser Art zu vermessen (Code-Metriken) sowie Testfälle zu erstellen. Zudem sollen typische Fehler von logischem Gestrüpp demonstriert und Beispiele von Code gezeigt werden, der nach state-of-the-art programmiert wurde. Schließlich sollen die Grenzen der hier vermittelten Metriken und Tests aufgezeigt werden.

AGENDA

Tag 1

I Theorie

- Einführung in die Grundlagen des Testens
- Problematik des Legacy Codes
- Grundbegriffe der Software-Metriken (Teil 1)

I Praxis

- Code-Metriken an Legacy Code
 - Grundbegriffe der Software-Metriken (Teil 2)
 - Praktische Einführung in die Codevermessung

Legacy Code: Wie teste ich chaotische Software? (Workshop)

Teil 2: Dynamischer Test

AGENDA

Tag 2

I Theorie

- Einführung in den dynamischen Test von Legacy Code

I Praxis

- Dynamischer Test von Legacy Code
 - Praktische Einführung in die Testfallerstellung mit Demonstration

VORKENNTNISSE

Die Teilnehmer*innen sollten praktische Kenntnisse in der Programmierung mit C/C++ haben. Vorkenntnisse im Testen sind nicht notwendig.

TECHNISCHE VORAUSSETZUNGEN

Eigener Laptop (alle technischen Details erhalten Sie in Ihrer Teilnahmebestätigung)

ZIELGRUPPE

Mittelständische Betriebe mit kleinem Entwicklungsteam; Betriebe mit älterer, gewachsener Software

Continuous Integration (Workshop)

Teil 1: Grundlagen CI

ABSTRACT

Das zweitägige Seminar befasst sich mit dem Prozess der fortlaufenden Integration von Komponenten zu einem System mit dem Ziel der Steigerung der Softwarequalität. Dabei wird typischerweise durchgehend überprüft, ob sich die Anwendung noch erstellen lässt. Um die Qualität der Software noch weiter zu erhöhen und beständig zu gewährleisten, sollten sich diesem Prozess noch weitere Schritte wie automatisierte Tests anschließen.

Gerade in Softwareprojekten mit mehreren Entwickler*innen kommt es immer wieder vor, dass nicht eingepflegte Dateien oder Änderungen an Komponenten eines Systems dazu führen, dass es sich nicht mehr erstellen lässt oder Teilbereiche nicht mehr funktionieren. Das kann ein großes Hindernis in der Weiterentwicklung des Projektes darstellen, wenn sich z.B. der/die verantwortliche Entwickler*in im Urlaub befindet.

Um dieser Problematik entgegenzuwirken, wurde der Prozess des Continuous Integration (CI) etabliert. Dabei dient ein Server gewissermaßen als weitere/r Mitarbeiter*in, der im einfachsten Fall nur dafür zuständig ist, nach jeder Änderung das Projekt neu zu erstellen.

Dieser praxisorientierte Workshop erläutert Ihnen die Grundlagen des Continuous Integration sowie auch Grundlagen des professionellen Softwaretests. Im praktischen Teil lernen Sie, wie Sie eine Versionsverwaltung am Beispiel Subversion aufsetzen. Anhand eines Beispielprojektes und mit Hilfe des Google Test Frameworks werden Sie in diesem Training verschiedene Arten des dynamischen Tests durchführen. Mit CppCheck lernen Sie ein praktisches statisches Analysewerkzeug kennen, mit dem Sie manche Fehler schon im Vorfeld erkennen können, ohne die Software ausgeführt zu haben. Unter Verwendung des CI-Servers Jenkins werden Sie alle erlernten Schritte zu einem automatisierten Prozess verbinden. Dabei werden sich Theorie und Praxisteil jeweils abwechseln.

Das Beispielprojekt besteht aus einer Server-Applikation, die sich auf dem BeagleBone Black befindet, und einem Client, der auf der Workstation läuft. Der Fokus dieses Seminars liegt auf der kontinuierlichen Integration der auf dem Embedded System befindlichen Server-Applikation. Dabei wird Ihnen u.a. gezeigt, wie man Hardware in the Loop (HiL) Tests automatisiert und mit Jenkins realisiert.

AGENDA

Tag 1

I Theorie

- Begriffserklärung Softwarequalität und Versionsverwaltung
- Was ist Continuous Integration?
- Warum Continuous Integration?
- Wie setze ich Continuous Integration ein?
- Grundlagen professioneller Softwaretest
 - Notwendigkeit des Testens
 - Was ist ein statischer Softwaretest und welche Arten gibt es?

I Praxis

- Vorstellung des Beispielprojektes und Vorbereitungen zu Continuous Integration
 - Aufsetzen eines VCS am Beispiel Subversion
 - Erstellung des Testframeworks Google Test
 - Projekterstellung und Cross-Compilation mit NetBeans
 - Probelauf des Beispielprojektes mit BeagleBone und Workstation
- Vorstellung des CI-Servers Jenkins
 - Automatisierung des Projekt-Buildprozesses mit Jenkins

Continuous Integration (Workshop)

Teil 2: Tests

AGENDA

Tag 2

I Theorie

- Dynamischer Softwaretest
 - Grundlagen und Begriffserklärung

I Praxis

- Statischer Test
 - Wie bereite ich einen statischen Test für CI vor (am Beispiel CppCheck)?
 - Durchlauf mehrerer CI-Zyklen mit statischen Tests
- Unittest
 - Vorbereitung, Erläuterung und Einstieg in die Übung Unittest
- Dynamischer Test und Einbindung CI
 - Fertigstellung der Übung zu Unittest
 - Erläuterung und Übung zu Integrationstest
 - Erläuterung und Übung zu Systemtest
- Gruppenübung CI
 - Erweiterung der Sourcen in Teamarbeit
 - Durchlauf möglichst mehrerer CI-Zyklen

VORKENNTNISSE

Die Teilnehmer*innen sollten praktische Kenntnisse in der Programmierung mit C/C++ haben. Vorkenntnisse im Testen sind nicht notwendig.

TECHNISCHE VORAUSSETZUNGEN

Eigener Laptop (alle technischen Details erhalten Sie in Ihrer Teilnahmebestätigung)

ZIELGRUPPE

Entwicklung, Projektmanagement

Continuous Integration: Nightly Build

ABSTRACT

Gerade bei größeren Softwareprojekten mit mehreren Entwickler*innen kommt es oft vor, dass die Weiterentwicklung des Projekts durch Dateien, die nicht in das Repository eingepflegt sind, erschwert wird oder sich das Projekt durch anderweitige Eingriffe nicht mehr erstellen lässt.

Nightly Build als Vorstufe der Continuous Integration sichert durch täglich nachts durchgeführte Erstellung von Projekten die stete Verfügbarkeit einer prinzipiell lauffähigen Software.

In dieser Expert Session werden die Grundprinzipien des Nightly Build sowie dessen Sinn und Zweck erläutert. Anhand eines Beispiels wird gezeigt, wie man mit Hilfe des Jenkins Build-Servers einen Nightly Build realisiert.

AGENDA

I Theorie

- Einführung
- Begriffsklärungen
- Nightly Build
 - Benefit
 - Funktionsweise
 - Voraussetzungen
- Übersicht über aktuelle Tools

I Praxis

- Demonstration der Funktionsweise automatisierter Builds mit Jenkins

VORKENNTNISSE

Interesse an Softwarequalitätsmanagement

ZIELGRUPPE

Entwicklung, Projektmanagement, Qualitätssicherung

IoT: Technologien und Entscheidungsgrundlagen für das Internet der Dinge (Workshop)

ABSTRACT

Sie kennen wesentliche Hardware- und Softwarekomponenten, Protokolle, Tools und deren Zusammenspiel, um Internet of Things (IoT) Architekturen und Industrie 4.0-Lösungen für Ihre Produkte zu entwickeln.

Mit dem hier vermittelten Themenüberblick entwickeln Sie die für Ihre Applikation passende Systemarchitektur und entscheiden fundiert über den Einsatz von Komponenten und Protokollen. Dabei kennen Sie in der Übersicht etablierte Konzepte zur Datensicherheit.

Die lauffähige Machine-to-Machine-(M2M) Kommunikation als Ergebnis der praktischen Übung ist für Sie gleichzeitig die Basis für Ihre weitere Produktevaluierung und Produktentwicklung. Dazu erhalten Sie die Gateway- und Sensorhardware, die verwendeten Softwarekomponenten und Entwicklungstools sowie die ablauffähigen Programmcodes.

AGENDA

- Einstieg
- IoT-Systemarchitekturen
- Hardwarekomponenten und Schnittstellen
- Einsatz von Betriebssystemen im IoT-Umfeld
- Kommunikationsprotokolle und Standards
- Webtechnologien und Frameworks
- Tools und deren Verwendung
- Datensicherheit in Embedded Systemen
- Einsatz von Clouds für Embedded Systeme
- Praktische Übung

VORKENNTNISSE

Grundkenntnisse über Mikrocontroller, Programmierkenntnisse in C

ZIELGRUPPE

Softwareentwickler*innen, Softwarearchitekt*innen, Hardwareentwickler*innen, Hardwarearchitekt*innen, Systemarchitekt*innen

Mein IoT-Gateway – von der abstrakten Architektur zur konkreten Implementierung

ABSTRACT

Im „Internet of Things“ kommunizieren verschiedenste Objekte miteinander, von der intelligenten Glühbirne bis hin zum vernetzten Auto.

Die Einsatzszenarien erstrecken sich vom „Smart Home“ bis hin zu intelligenten Industrieanlagen. Ziel des Vortrags ist das Veranschaulichen verschiedener Architekturansätze sowie das Vorstellen konkreter IoT-Technologien. Hierzu werden im ersten Teil mehrere Architekturen aus verschiedenen Bereichen veranschaulicht. Darauf aufbauend werden mögliche Komponenten und Technologien, mit denen ein IoT-Gateway realisiert werden kann, vorgestellt. Zum Abschluss wird ein Ausblick in dem Bereich Security gegeben.

AGENDA

- Einführung – was ist IoT?
- Architektur von IoT
- Mein IoT-Gateway
 - Aufbau und Architektur
 - Vorstellung einiger Funkprotokolle
 - Mögliche Kommunikationsprotokolle
- Ausblick: Security im IoT-Gateway

VORKENNTNISSE

Technisches Verständnis

ZIELGRUPPE

Entwicklung, Architektur

Maschinensteuerung mit Web-Technologien

ABSTRACT

Web-Oberflächen haben nicht nur den Vorteil, dass sie flexibel und betriebssystemübergreifend einsetzbar sind – sie bieten durch ihre große Verbreitung auch viele Hilfsmittel, mit denen unter angemessenem Aufwand eine intuitiv bedienbare, moderne und ästhetisch ansprechende Oberfläche gestaltet werden kann.

Zur Realisierung von Maschinensteuerungen über Webseiten sind Aufgaben zu meistern, die in den traditionellen Gebieten des Internets selten zu finden sind, wie beispielsweise die Kommunikation mit der zu steuernden Hardware.

Das Webinar beschäftigt sich mit diesem Thema genauer, indem die Umsetzungsmöglichkeit anhand eines Sortierers für LEGO® Steine aufgezeigt wird, der im Rahmen einer Masterarbeit entstanden ist.

AGENDA

- Vorteile grafischer Nutzerschnittstellen
- Vorteile webbasierter Oberflächen
- Vorstellung des LEGO® Stein-Sortierers
- Umsetzung der Nutzersteuerung
- Webspezifische Eigenheiten

VORKENNTNISSE

Interesse am Thema „Web-Technologien“

TECHNISCHE VORAUSSETZUNGEN

Eigener Laptop (alle technischen Details erhalten Sie in Ihrer Teilnahmebestätigung)

ZIELGRUPPE

Entwicklung und Entscheider*innen, die einen Einstieg in das Thema Web-Technologien suchen

EXPERT SESSIONS

© 2022 Mixed Mode GmbH - an Ingenics company



Mixed Mode GmbH
Lochhamer Schlag 17
D-82166 Gräfelfing

Tel.: +49/89/8 98 68-200
Fax: +49/89/8 98 68-222

seminar@mixed-mode.de
www.mixed-mode.de