

2 Ausgangssituation und Zielsetzung

In diesem Thema werden die Ausgangssituation und der Anlass zur Durchführung des Projekts anschaulich dargestellt. Es wird beschrieben, welche Defizite bzw. Probleme existierender Systeme oder auch der aktuellen Situation zur Entscheidung geführt haben, das Projekt durchzuführen, und welche Vorteile durch den Einsatz des neuen Systems erwartet werden. Zusätzlich werden alle relevanten Stakeholder des Projekts benannt und die technische und fachliche Einbettung des zu entwickelnden Systems in seine Umgebung skizziert. Zusätzlich werden erste Rahmenbedingungen für die Entwicklung identifiziert und beschrieben. Rahmenbedingungen können beispielsweise technische Vorgaben oder Vorgaben zur Sicherheit sein.

Jedes Jahr wieder müssen Schüler, die im letzten Jahreszeugnis eine negative Beurteilung im Zeugnis hatten zu Wiederholungsprüfungen eingeteilt werden. Dies geschieht durch „Zusammenbauen“ des Prüfungsplanes mittels Kärtchen. Diese werden so lange umher geschoben und umgelegt, bis es sich sozusagen endlich ausgeht alle unter zu bringen.

Bei dieser derzeitigen Ausgangslage muss das Planungskomitee alle Planungsaufgaben per Hand erledigen. Dies ist nicht mehr zeitgemäß und auch dermaßen umständlich, dass immer wieder Fehler auftreten, welche dann durch kurzfristige Improvisation kompensiert werden müssen.

Mit diesen Improvisationen ist meist jede vorhergehende Organisation hinfällig oder wenigstens durcheinander geworfen. Somit kommt es leider vor, dass der ohnehin schon belastende Prüfungsstress sinnlos erhöht wird, und dies ist natürlich angesichts der Wichtigkeit der Prüfung für Prüfer und Prüflinge nicht optimal.

Das Projektziel ist eine Automatische Terminplanung für Wiederholungsprüfungen der HTBLA Leonding inklusive zeitliche und gegebenenfalls räumliche Kollisionsprüfungen. Der ganzen Terminplanung sollte ein individuell gestaltbares und leicht veränderbares Regelwerk zugrunde liegen, aus dem dann die Planungsparameter entnommen werden. Anhand dieser Regeln soll dann automatisch ein Plan der Wiederholungsprüfungen generiert werden.

H. Kov. d. Proj. - Vorschlag

3 Funktionale Anforderungen

Funktionale Anforderungen beschreiben die Fähigkeiten eines Systems, die ein Anwender erwartet, um mit Hilfe des Systems ein fachliches Problem zu lösen. Die Anforderungen werden aus den zu unterstützenden Geschäftsprozessen und den Ablaufbeschreibungen zur Nutzung des Systems abgeleitet. Beschreibung der funktionalen Anforderungen erfolgt beispielsweise in Form von Anwendungsfällen (Use Cases). Ein Anwendungsfall beschreibt dabei einen konkreten, fachlich in sich geschlossenen Teilvorgang. Die Gesamtheit der Anwendungsfälle definiert das Systemverhalten. Ein Anwendungsfall kann in einfachem Textformat beschrieben werden, häufig stehen jedoch organisationsspezifische Muster zur Beschreibung zur Verfügung. Für datenzentrierte Systeme wird im Rahmen der funktionalen Anforderungen ein erstes fachliches Datenmodell erstellt, das als Grundlage des späteren Datenbankentwurfs dient. Das fachliche Datenmodell des Systems wird aus den Entitäten des Domänenmodells abgeleitet. Funktionalen Anforderungen sind die zentralen Vorgaben für die Systementwicklung. Sie werden in der Gesamtsystemspezifikation (Pflichtenheft) übernommen und bei Bedarf konkretisiert.

Die funktionalen Anforderungen sind:

- **Regelwerk:**
 - Eingabemöglichkeit der Regeln zum Beispiel in Form eines Editors, eines Formulars, etc. Genauerer kann noch nicht gesagt werden, da die Form/Art (Parameter) der Regeln noch nicht feststehen.
 - Änderungsmöglichkeit der Regeln kann wieder mit einem Editor oder einem Formular, etc. realisiert werden. Auch hier kann noch nichts Genauerer gesagt werden.
 - Syntaxprüfung der Regel: Nach der Eingabe der Regeln müssen sie auf die syntaktische Richtigkeit kontrolliert werden
- **Manuelles Eingreifen des Benutzers in die Terminplanung**
 - Dieses Eingreifen soll durch eine einfache, verständliche aber auch leistungsfähige Grafikoberfläche unterstützt werden, bei der der Benutzer leicht Änderungen vornehmen kann. Ein Bsp. Dafür wäre das fixe Setzen von Terminen
- **Kollisionsprüfungen:**
 - Manuelle Kollisionsprüfung, die der User selbst auslösen kann um den Plan auf Kollisionen zu überprüfen
 - Automatische Kollisionsprüfung: Dabei sind 2 Arten angedacht:
 1. Bei der automatischen Erstellung des Planes kontrolliert sie und meldet Fehler die dann automatisch ausgebessert werden.
 2. Bei einer manuellen Änderung des Planes durch den Benutzer soll/kann eine automatische Kollisionsprüfung, die im Hintergrund läuft, die Änderungen kontrollieren. Doch es ist aus Performancegründen noch genau klar wie und ob er realisiert wird. Vorstellbar wäre, dass immer in regelmäßigen Abständen kontrolliert wird. Ein vermutlich besserer Vorschlag wäre das explizite Auslösen der Kollisionsprüfung durch den User.
 - Ausschalten der Kollisionsprüfung ganz oder auf einzelne Elemente soll ebenfalls möglich sein, da Kollisionen vereinzelt auch gewollt sein können (z.B.: wegen wechselseitiger Rollen als Prüfer und Zweitprüfer)

Anwendungsfall „Terminplanung“:

Die folgende Situation wird unseren primären Hauptanwendungsfall darstellen:

Alle Schüler die eine Wiederholungsprüfung zu absolvieren haben stehen fest. Nun macht sich ein Team von Lehrern daran eben diese Prüfungen zu planen und einzuteilen. Dies geschieht derzeit mit einem unübersichtlichen „Terminplan“: Schüler, Lehrer und Räume sind auf Kärtchen gedruckt und werden auf diesem Plan solange umgelegt bis sich alle Prüfungen ausgegangen sind. Dies ist ein großer Aufwand und sehr zeitintensiv, nebenbei kann man leicht einen Fehler übersehen.

An dieser Stelle hilft das Automatische Terminplanungssystem für Wiederholungsprüfung.

Nach dem Importieren des Excel-CSV-Files mit allen Prüfungen erstellt es automatisch einen Terminplan, welcher nachher von einem Prüfer, der gegebenenfalls manuell in den erstellten Plan eingreifen kann, durchgesehen wird. Auf Wunsch des Auftraggebers kann der fertige Plan auch noch in die bereits vorhandene Access-Datenbank importiert werden.

4 Nicht-funktionale Anforderungen

Nicht-funktionale Anforderungen beschreiben Anforderungen an das System, die nicht-fachlicher Natur sind, jedoch entscheidend zur Anwendbarkeit des Systems beitragen. Sie definieren beispielsweise Qualitätsanforderungen, Sicherheitsanforderungen oder Performanceanforderungen. Funktionale Anforderungen definieren grundlegende Eigenschaften eines Systems, die im Architekturentwurf berücksichtigt werden müssen. Sie können zur Abschätzung der Entwicklungskosten herangezogen werden und sollten, soweit möglich, messbar beschrieben sein. einfachen Strukturierung der Anforderungen werden diejenigen Anforderungen, die nicht eindeutig zu den funktionalen Anforderungen gehören, den nicht-funktionalen Anforderungen zugeordnet.

- * Grundlegend wichtig für die Qualität, die Performance und ferner auch für die Sicherheit des Systems ist jene CSV-Eingabedatei, welche die Auflistung aller Prüfungen liefert. Sollte diese nicht stimmen, oder deren Format inkorrekt sein ist es denkbar dass das komplette System falsche Werte liefert oder schlechtesten falls überhaupt nicht funktioniert.
Durch die enthaltenen geprüften Regelsätze soll die Qualität des Ergebnisses konstant hoch gehalten werden können. Weitere Sicherheitsanforderungen sind für die schulische Verwendung gänzlich unnötig, da die Software von außen nicht beeinflussbar sein wird.
- * Hinsichtlich der Berechnungsperformance für die Termine müssen genannte Regelsätze performance-optimal definiert und auch auf dieses Merkmal besonders getestet werden.
Um den Abschluss inklusive Export in die vorhandene Access-Datenbank erfolgreich automatisch durchführen zu können dürfen natürlich an diesem DB-System keine (grundlegenden) Änderungen vorgenommen werden bzw. der Export muss dem angepasst werden. Microsoft Access stellt dazu bereits genügend vordefinierte Funktionen zur Verfügung.
- * Aufgrund der Modifizierbarkeit des Regelwerks ist besonders zu beachten, dass erstens die Performance nicht all zu sehr leidet (WorstCase wäre die Berechnungsunfähigkeit durch widersprüchliche
- * Regelangaben oder Ähnliches), zweitens darf die Stabilität nicht durch Benutzereingriffe gefährdet
- * werden und drittens soll der Programmstrukturaufbau trotz allem einfach bleiben, um die Bedienbarkeit möglichst einfach zu gestalten.

"Roman" (Hilfere Aufzählungssätze)

NANF 1

2

6 Lebenszyklusanalyse und Gesamtsystemarchitektur

Ausgehend von den Anforderungen werden ein grober Entwurf des Gesamtsystems erstellt und die zu unterstützenden Phasen im Lebenszyklus (Entwicklung, Wartung, Stilllegung) identifiziert. Der Gesamtsystemarchitektur wird das zentrale System mit den Unterstützungssystemen identifiziert und festgelegt, für welche Systeme ein logistisches Unterstützungskonzept zu erstellen ist. Grundlage sind die funktionalen und nicht-funktionalen Anforderungen sowie die Skizze der Gesamtsystemarchitektur in den Anforderungen. Beistellungen des Auftraggebers werden berücksichtigt. Gesamtsystemarchitektur wird hinsichtlich der möglichen Verwendung von Externen Einheiten geprüft. Für alle identifizierten Externen Einheiten wird eine Make-or-Buy-Entscheidung herbeigeführt, gegebenenfalls unterstützt durch eine Marktsichtung geeigneter Kandidaten.

Das Gesamtsystem besteht aus dem Regelwerk, der Grafikoberfläche, dem Planungsteil und den Kollisionsprüfungen, des Weiteren besteht es aus Schnittstellen, die die CSV-Datei einlesen, bzw. das Ergebnis in die Access-Datenbank schreiben.

Welche Anforderungen die einzelnen Teile des Gesamtsystems erfüllen sollen wurde bereits in den Kapiteln 3 (funktionale Anforderungen) bzw. 4 (nicht-funktionale Anforderungen) geklärt.

Die einzelnen Phasen des Lebenszyklus, Entwicklung, Wartung, Stilllegung werden größtenteils von externen Personen, d.h. von Personen die an der Entwicklung des Projektes nicht maßgeblich beteiligt sind, abgearbeitet. Dies ergibt sich daraus, da die Mitglieder des Projektteams nach Abschluss der Matura nicht mehr an der Schule sind und somit die Phasen Wartung und Stilllegung nicht übernehmen können.

Wer diese Phasen überwachen bzw. durchführen will steht derzeit noch nicht fest.

siehe Fort.

7 Schnittstellenübersicht

Zur Darstellung der Zusammenhänge zwischen dem System und seiner Umgebung wird eine Schnittstellenübersicht erstellt. Ausgehend vom System werden Schnittstellen zum Anwender, zu den Unterstützungssystemen, zur Logistik und zu Nachbarsystemen identifiziert und in geeigneter Form dokumentiert. Konkrete Beschreibung der Schnittstellen erfolgt in den Spezifikationen der Systemelemente sowie in der Spezifikation logistische Unterstützung.

MS Excel-Datei

Die CSV-Datei wird aus dem Schulverwaltungssystem (SVS) generiert. Diese Datei dient dem System als Eingabe, sie enthält alle Lehrer und Schüler, die an den Prüfungen beteiligt sind. Der Inhalt dieser Datei ist nach den Lehrern sortiert, so dass diese vom System abgefragt werden können.

Oberfläche

Die Oberfläche dient dazu den erstellten Prüfungsplan anschaulich darzustellen. Außerdem soll es dem Benutzer mit Hilfe dieser Oberfläche möglich gemacht werden, dass er auch nach der Berechnung des Planes noch manuelle Änderungen vornehmen kann. Natürlich werden mit der Oberfläche auch noch eine Reihe von Funktionen zur Verfügung gestellt wie Speichern, Einstellungen, Exportieren, uvm.

Access-Datenbank

Der fertige Plan soll nach Bestätigung des Benutzers in eine Access-Datenbank geschrieben werden, diese Access-Datenbank existiert bereits und ist derzeit auch schon in Verwendung. Für diese Datenbank sind auch schon verschiedenste Reports geschrieben worden um den fertigen Terminplan auf verschiedene Weisen ausdrucken zu können.

