

Praktikumsaufgabe Software Engineering¹

Ablauf des Praktikums

Das Praktikum besteht aus 6 Terminen. In den Terminen 1-5 wird die Software zu einer **Cocktailmischanlage** „CocktailPro“ von der Anforderungsanalyse bis zum fertigen Programm entwickelt (Natürlich stark vereinfacht, ohne Fehler- und Ausnahmebehandlung, ohne GUI, ohne Datenbankbindung, ohne Anbindung an eine reale Cocktailmischanlage, ohne Ressourcenbeschränkung). Im Termin 6 wird die Analyse der in C++ geschriebenen Anwendungen mit Hilfe von Metriken durchgeführt.

Das Praktikum setzt voraus, dass Sie mit den Grundlagen der Modellierung und der Bedienung des Case-Tools MagicDraw vertraut sind. Ist dies nicht der Fall, sollten Sie die Übung aus OOAD zur Auffrischung noch einmal durcharbeiten und sich die Anleitung für MagicDraw unter <https://www.fbi.h-da.de/labore/case/laborausstattung/magicdraw/einfuehrung-magicdraw.html> durchlesen.

In dem Praktikum besteht Anwesenheitspflicht. Unentschuldigtes Fernbleiben führt zum Ausschluss und Nichtbestehen des Praktikums. Die im jeweiligen Praktikum durchzuführenden Tätigkeiten sind vor dem Praktikumstermin vorzubereiten.

Falls eine Gruppe unvorbereitet zum Praktikum erscheint oder zu Beginn des darauf folgenden Praktikums nicht die zum vorherigen Termin geforderte Leistung erbracht hat, wird einmalig ermahnt (GELBE KARTE). Beim 2. Vorkommen dieser Art wird der Studierende vom weiteren Besuch der Veranstaltung ausgeschlossen und das Praktikum als nicht bestanden gewertet (ROTE KARTE). Eine Zulassung zum Leistungsnachweis ist dann nicht möglich, das Praktikum muss in einem folgenden Semester wiederholt werden.

Warnungen

- In diesem Praktikum geht es um Software Engineering. Deshalb stehen das Vorgehen, Modelle, Entwurfsmuster etc. im Vordergrund der Aufgaben. Dass am Schluss ein lauffähiges Programm entsteht, das die Anforderungen erfüllt, wird als selbstverständlich angesehen. Für die Endabnahme müssen Sie demonstrieren, dass Ihr Modell und Ihr Code in Einklang stehen.
- In diesem Praktikum arbeiten Sie weitgehend selbständig und werden nur wenig überwacht. Insbesondere ist die Abnahme für die ersten Termine wesentlich freizügiger als z.B. in OOAD. Sie könnten die Abnahme für die Anforderungsanalyse prinzipiell auch erhalten, wenn Sie einen einzigen Use Case modellieren, der einen Cocktail mischt. Bei der Klassenmodellierung werden Sie dann allerdings feststellen, dass Sie die Anforderungen nicht hinreichend genau kennen und die Arbeit dort nachholen. Das gleiche passiert, wenn Sie Sequenzdiagramme vernachlässigen und sofort mit der Entwicklung starten. Das Vernachlässigen der Ergebnisse einer Phase führt erfahrungsgemäß zu deutlich höherem Aufwand in den folgenden Phasen ("10-er Regel", weil sich der Aufwand verzehnfacht). Das ist zwar sehr ineffizient aber dafür umso lehrreicher...
- Die Implementierung erfordert die Beherrschung von C++. Sie sollten jedenfalls mit Klassen, Vererbung, Konstruktoren und Pointern vertraut sein. Ansonsten ist das Praktikum kaum zu bestehen.

Nutzung von MagicDraw

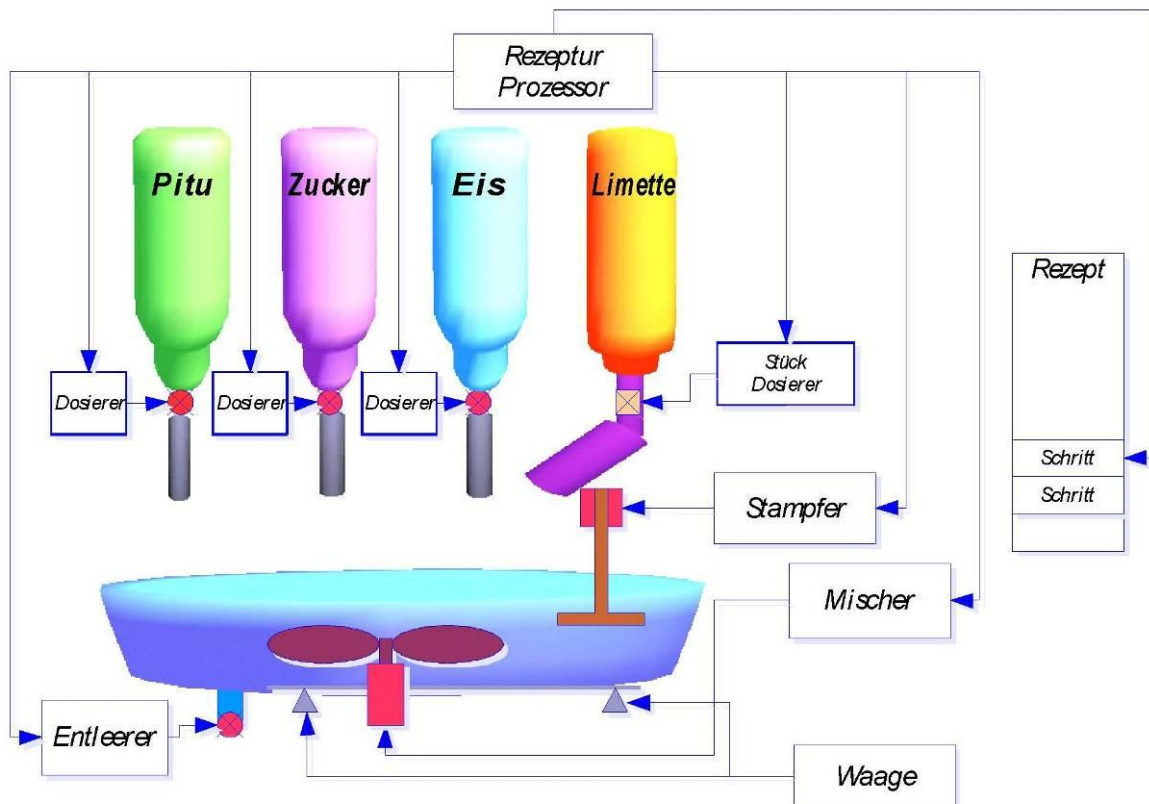
Die Bedienung von MagicDraw wurde in OOAD bereits geübt. Zur Klärung von Fragen zur Bedienung ist es sinnvoll, die Anleitung für MagicDraw vorliegen zu haben <https://www.fbi.h-da.de/labore/case/laborausstattung/magicdraw.html>

¹ Die Aufgabenstellung wurde von Prof. Dr. Ralf Hahn und Michael Guist entwickelt. Die Nutzung im Sommersemester 2014 erfolgt mit ausdrücklicher Genehmigung der Autoren.

Aufgabe des Gesamtsystems CocktailPro

Entwickeln Sie ein System zum automatisierten Zubereiten von Cocktails. Ein Benutzer wählt ein Rezept aus. Anschließend werden die in den Rezept-Schritten angegebenen Zutaten (z.B. Pitu, Zucker, Eis und Limettenstücke für einen Caipirinha) in der richtigen Menge und Reihenfolge in einen Behälter gefüllt, gemäß Vorgabe im Rezept für eine definierte Zeit gestampft, gerührt und anschließend entleert.

Der schematische Aufbau der technischen Anlage ist in der folgenden Abbildung dargestellt.



Für jede Zutat ist eine Dosierstation vorhanden. Ein Dosierer liefert auf ein Kommando hin eine bestimmte Menge der Zutat. Ein Stückdosierer liefert eine abgezählte Anzahl von Stücken (z.B. Limetten).

Die dosierten Zutaten werden über Rohre in einen Mischbehälter gefüllt. Im Mischbehälter kann die Mischung gerührt oder gestampft werden. Die Stampf- bzw. Rührereinrichtung kann für eine Zeitdauer aktiviert werden. Der Mischbehälter besitzt ein Entleerventil, mit dem der Cocktail entleert werden kann. Um den Dosiervorgang zu überwachen, steht der Mischbehälter auf einer Waage, mit der der Inhalt gewogen wird. Der Mischbehälter besitzt eine Vorrichtung zum automatischen Reinigen.

Die Dosierstation

- Eine Dosierstation besitzt jeweils ein Ventil das geöffnet und geschlossen werden kann. Die Menge der bereits dosierten Zutat muss über die Waage kontrolliert werden.
- Der Stückdosierer funktioniert genauso wie die „normalen“ Dosierer.
- Durch die Befüllung der Dosierstationen mit Zutaten werden die mischbaren Rezepte bestimmt (in der Regel sind nicht alle Rezepte des Rezeptbuchs mischbar).

Der Rezepturprozessor

- Die Verarbeitungsschritte sind in einem Rezept (s. u.) festgelegt.
- Der Rezepturprozessor führt, falls notwendig, koordinierende Tätigkeiten aus.

Das Rezept

- Das Rezept ist eine geordnete Folge von Rezeptschritten, die beim Herstellen des Getränks durchgeführt werden müssen.

Die Rezeptschritte

- Die Rezeptschritte sind die einzelnen Schritte des Rezepts.
- Jeder Rezeptschritt enthält die Information welche Station den Schritt ausführen soll und den Parameter, den sie dazu benötigt (Menge / Anzahl / Zeitdauer).

Das Rezeptbuch (EXTERNE ZULIEFERUNG!!!)

- Das Rezeptbuch entspricht einem Buch der Art "Die 1000 Cocktails der Welt" und verwaltet eine Menge von Rezepten und den dafür benötigten Zutaten und Dosierungen
- Es liefert eine nummerierte Liste mit Rezepten
- Zu einem (über die Nummer) ausgewählten Rezept kann der i-te Rezeptschritt abgefragt werden. Es wird die erforderliche Zutat (bzw. der Verarbeitungsschritt) und die Menge bzw. Verarbeitungsdauer geliefert.
- Für die Entwicklung erhalten Sie eine vorläufige Version des Rezeptbuchs, das nur ein paar Rezepte enthält.

Die Waage

- Die Waage liefert 2 Gewichtswerte. Das absolut gewogene Gewicht und ein Differenzgewicht „Delta“, das relativ zu einem frei setzbaren Delta-Nullpunkt ist.
- Die Waage ist so geeicht, dass das Absolutgewicht mit dem leeren Mischbehälter 0 ist.
- Der „Delta-Nullpunkt“ kann durch einen Aufruf auf das aktuelle Absolutgewicht gesetzt werden

Der Mischer

- Der Mischer mischt für eine definierte Zeitdauer.

Der Stampfer

- Der Stampfer stampft für eine definierte Zeitdauer

Der Entleerer des Mischbehälters

- Der Mischbehälter verfügt über ein Ventil zur Entleerung („Entleerer“) das geöffnet und geschlossen werden kann. Die Restmenge im Mischbehälter muss über eine Waage kontrolliert werden.
- Der Mischbehälter reinigt sich automatisch nach der vollständigen Entleerung

Das Display

- Das Display bietet die Cocktails zur Auswahl und gibt Informationen zum Mischvorgang aus.

Szenario

Mischen eines Cocktails

1. Systemstart.
2. Die (mit den vorhandenen Zutaten mischbaren) Cocktails werden als Auswahl angeboten.
3. Der Benutzer wählt einen Cocktail aus.
4. Entsprechend der Rezeptvorgaben werden Kommandos an Dosierer geschickt.
5. Das Ventil des Dosierers X wird geöffnet.

6. Wenn die Waage das gewünschte Zusatzgewicht (Delta) erreicht oder überschritten hat, wird das Ventil des Dosierers X geschlossen.
7. Analog zu Schritt 4., 5. und 6. werden die anderen Zutaten dosiert.
8. Der Mixer und/oder Stampfer wird für eine bestimmte Zeit aktiviert.
9. Der Cocktail wird entleert.
10. Die Reinigung des Mischbehälters erfolgt.
11. Goto 2.

Randbedingungen

- Das Rezeptbuch liegt bereits vor - als Binärkomponente mit C++ Header, einer Beispiel-Anwendung (main.cpp) und der Dokumentation des zugehörigen MagicDraw-Modells.
- Es wird Quellcode zum Einlesen einer Datei bereit gestellt.
- Es wird Quellcode zum Warten von x msec. bereit gestellt.
- Es wird Quellcode bereit gestellt, der die Verwendung der Standardbibliothek <list> zeigt
- Es wird eine Vorgängerversion als Demo bereit gestellt. Sie hat zwar nicht vollständig die gewünschte Funktionalität, aber sie veranschaulicht das Verhalten des Systems.

Aufgabenstellung

In der vorhergehenden Beschreibung der zu realisierenden Anlage hat der Kunde aus seiner Sicht und mit seinen Beschreibungsmitteln die Funktionsweise der Anlage beschrieben.

Wir wollen nun diese Beschreibung umsetzen in eine präzise mit UML beschriebene Spezifikation, die dann anschließend mit dem Kunden durchgesprochen wird (bei uns simuliert durch ein Anforderungs-Review, das durch eine anderer Praktikumsgruppe durchgeführt wird).

Im Rahmen des Praktikums wird ein Prototyp des Systems entwickelt. Dieser verwendet Software-Simulationen der Geräte wie Dosierer etc. und läuft komplett auf einem Standard-Rechner.

Der Prototyp soll so realitätsnahe wie möglich entwickelt werden. Dazu wird die fehlende Hardware durch Software-Simulationen ersetzt, die Sie u. a. selbst entwickeln sollen. (Das heißt, anstatt eine echte Waage anzusteuern, simulieren Sie diese mit Software.

Die Simulations-Funktionen simulieren die fehlende Hardware und liefern die Informationen, wie sie von der echten Hardware erwartet würden oder die Simulations-Funktionen nehmen die Informationen an und geben sie am Bildschirm aus, statt sie an die Hardware (z. B. Anzeigegeräte) weiterzugeben. Durch die Verwendung dieser Simulations-Funktionen kann das eigentliche Mixen der Cocktails in der Simulation genauso ablaufen wie auf einem echten Zielsystem. Auf dem Zielsystem müssten später „nur noch“ die Simulations-Funktionen durch Funktionen mit gleichem Namen und Interface ersetzt werden - die dann allerdings auf die echte Hardware zugreifen.

Vorgaben:

1. Der Prototyp enthält 10 Dosierer + 1 Mixer + 1 Stampfer
2. Die im Prototyp nicht vorhandene Hardware (Dosierer, Waage usw.) wird durch entsprechende Funktionen simuliert,
3. Ausgaben von simulierten physischen Geräten (z.B. Waage) sollen über Bildschirmausgaben (cout) sichtbar gemacht werden.
4. Sämtliche Aktionen von Geräten werden durch entsprechende Bildschirmausgaben protokolliert (z.B. „Ventil von Dosierer O-Saft geöffnet“).
5. Auf dem Display der Waage wird sowohl das absolute als auch das Delta-Gewicht angezeigt
6. Solange ein Füllventil eines Dosierers geöffnet ist, soll sich das Gewicht auf der Waage selbstständig erhöhen.
 - Werden Limettenstücke dosiert, fällt jede Sekunde ein Stück auf die Waage. Simulieren Sie so, dass jedes Stück 10g wiegt.
 - Eis wird nach Gewicht dosiert. Jede Sekunde wird ein Eiswürfel dosiert. Simulieren Sie, dies mit 20g pro Eiswürfel - Sie können nur Vielfache von 20g genau dosieren.
 - bei Flüssigkeiten und feinkörnigen Zutaten (z.B. Zucker) ändert sich das Gewicht gleichmäßig. Simulieren Sie dies mit 1g/0,25s.
 - Ist das Entleerventil geöffnet, so verringert sich das Gewicht der Waage schnell. Simulieren Sie dies durch eine Reduzierung um 25g/s bis der Wert Null erreicht ist.
7. Das Abwarten von Zeit wird durch ein Wait-Statement simuliert. Am Bildschirm wird pro Sekunde ein '*' ausgegeben. Zu Testzwecken ist es möglich, die Wartezeit zu „kürzen“ (d.h. eine Sekunde wird durch eine 10-tel Sekunde simuliert).
8. Der Wirt soll konfigurieren können, welche Zutat in welchem Dosierer enthalten ist. Während des Systemstarts wird eine Text-Datei „zutaten.txt“ über ein Programm ähnlich dem zugelieferten Programm „readfile.cpp“ eingelesen, welche die Zuordnung von Dosierern zu Zutaten festlegt (Datei wird bereitgestellt). Dadurch ist eine Änderung der Zuordnung von Dosierern zu Zutaten ohne Neuübersetzung des Programms möglich.
9. Das System soll nach dem Start eine Auswahl-Liste der mischbaren Cocktails anzeigen (Im Prototypen soll auch die Konfiguration (Dosierer: Zutat) ausgegeben werden). Es werden nur die Cocktails zur Auswahl angeboten, für die auch alle Zutaten vorhanden sind.
10. Ausnahmen werden nicht realisiert (Dosierer hängt, ist leer etc.).

11. Um zukünftige Erweiterungen zu erleichtern, soll die Kommunikation zwischen der Waage und den Beobachtern der Waage (Dosierer, Mischbehälter und Display (der Waage)) nach dem Beobachter-Muster erfolgen.

Hinweise

- Das System soll hinter dem Tresen installiert werden und wird nur vom Personal bedient. Im Prinzip erfolgt die Bedienung genau wie bei einer Espressomaschine in einem Restaurant.
- Die vorhergehende Beschreibung des Systems gibt eine mögliche Aufteilung in Klassen (aus Sicht des Auftraggebers) vor. Prüfen Sie eingehend, ob diese Klassen wirklich sinnvoll sind.
- Analysieren Sie die Rezepte und deren Inhalt genau. Gibt es logische Unterschiede bei den Zutaten, die Sie auch entsprechend modellieren sollten?
- Eine der kniffligsten Stellen im gesamten System ist die Interaktion zwischen den Dosierern / Display und der Waage. Da das Gewicht nicht tatsächlich erhöht wird, muss die Gewichtserhöhung simuliert werden. Überlegen Sie genau, wo und wie Sie diese Funktionalität umsetzen.
- Überlegen Sie sich wie der Systemstart ablaufen soll und wie die Konfiguration eingelesen wird.
- Denken Sie an die Entwurfs-Grundprinzipien (Geheimnisprinzip, starke Kohäsion, schwache Kopplung), um einen möglichst stabilen Entwurf zu erhalten. Denken Sie auch an naheliegende Änderungen im System (insbesondere Änderungen der Rezepte und Zutaten).
- Schauen Sie sich die Schnittstellen des zugelieferten Rezeptbuchs an, bevor Sie mit Ihrem Entwurf beginnen.
- Es wird Source-Code („readfile.cpp“) zum Einlesen der Datei „zutaten.txt“ bereitgestellt, den Sie verstehen und für Ihren Bedarf modifizieren sollen.
- Es ist oft sinnvoll Standard-Datencontainer wie z.B. LIST zu verwenden. Ein Code-Beispiel für die Verwendung von LIST wird bereitgestellt.
- In der Endabnahme wird eine veränderte Zutatendatei eingespielt. Ihr System muss diese einlesen, dann die mischbaren Cocktails im Auswahlmenü anzeigen und 1-2 Cocktails richtig mischen. Dabei wird auf die Einhaltung der vorgegebenen Stückelung (vgl. Vorgabe 6 auf Seite 5) und Geschwindigkeit geachtet.
- In der Endabnahme wird ein Cocktail im normalen langsamen Mischmodus gemischt und ein Cocktail im beschleunigten Modus (10-fache Geschwindigkeit vgl. Vorgabe 7 auf Seite 5). Es wird Source-Code („Timer.cpp“) zum Abwarten einer definierten Zeitspanne bereitgestellt, den Sie verstehen und für Ihren Bedarf modifizieren sollen.
- Es gibt für dieses Praktikum viele richtige Lösungen. Ihre Aufgabe ist es – so wie in einem echten Projekt auch – einen geordneten und sicheren Weg zu Ihrer Lösung zu finden. Sie sollen während des Praktikums ein Gefühl dafür bekommen, wann eine Entwicklungsphase hinreichend gründlich bearbeitet wurde.

Hinweise zur Technik

- Textdateien (z.B. .csv oder .txt) unter Windows und Linux haben ein unterschiedliches Dateiformat und werden von manchen Editoren beim Speichern automatisch in das entsprechende Format des Betriebssystems gewandelt. Wenn Sie also die Datei „zutaten.txt“ unter Windows anschauen und speichern und anschließend die Datei mit Ihrem Code nach Linux kopieren, liegt dort das Windows-Format vor und es kommt beim Einlesen der Texte zu seltsamen Effekten (Strings sind leer etc.). Die Unixbefehle dos2unix bzw. unix2dos wandeln die Dateiformate. Unter Windows bieten Editoren wie z.B. Notepad++ die Möglichkeit zur Auswahl des Speicherformats.
- Wenn Sie eine Folge von Sternen o.ä. in einer Zeile ausgeben wollen, müssen Sie bei ANSI-C++ den Ausgabepuffer nach jeder Ausgabe mit `cout << << flush` explizit leeren.

Aufgaben

1. Termin: Anforderungsanalyse

Vorbereitung

- Erstellen Sie den Use Case "Cocktail mischen" mit einer textuellen Beschreibung auf einem Blatt Papier. Dazu frischen Sie Ihr Wissen auf², über die textuelle Beschreibung von Use Cases und UML Use Case Diagrammen.
- Arbeiten Sie die vorangehende Systembeschreibung und die Hinweise genau durch. Dokumentieren Sie aufkommende Fragen.
- Hinweis: Das Praktikum setzt voraus, dass Sie mit der Bedienung des Case-Tools MagicDraw vertraut sind. Ist dies nicht (mehr) der Fall, sollten Sie die Übung aus OOAD zur Auffrischung noch einmal durcharbeiten und in der Anleitung zu MagicDraw den Abschnitt zu den Use Case-Diagrammen lesen.

1.1.) Anwendungsfälle (Use-Cases)

Analysieren Sie die vorhergehende Systembeschreibung auf Anwendungsfälle, die das System erfüllen muss. Erstellen Sie ein Use-Case-Diagramm und beschreiben Sie die einzelnen Use-Cases im Modell.

1.2.) Einbinden des Rezeptbuchs

Das Rezeptbuch wird von Extern zugeliefert³.

Der Betreuer erklärt Ihnen, wie Sie die Zulieferung in Ihr Modell integrieren. Zum Kompilieren des Anwendungsbeispiels beachten Sie die Hinweise im Anhang A auf Seite 12.

1.3.) Anforderungs-Review

Der Betreuer zeigt Ihnen, wie Sie ein Dokument mit Ihren Arbeitsergebnissen erzeugen. Prüfen Sie den Inhalt und drucken Sie dann das Dokument aus.

Setzen Sie sich mit einer anderen Gruppe zusammen und führen Sie gegenseitig ein Review durch. Prüfen Sie, ob die Use-Case-Spezifikation den in der Aufgabenstellung aufgeführten Angaben entspricht. Verwenden Sie dabei die bereitgestellten Review-Vorlagen.

Ergebnis

Sie haben genau verstanden und dokumentiert, was die Aufgabe des CocktailPro ist. Unklarheiten haben Sie herausgearbeitet und durch Rückfragen geklärt. Die Qualität der Anforderungen wurde durch ein Review bestätigt.

Ihr Modell enthält ein Use Case Diagramm und mehrere sinnvolle und konsistente Use Cases. Die Spezifikationen sind ausgefüllt. Das zugelieferte Rezeptbuch ist in Ihr Modell eingebunden.

2. Termin: Design – der erste Entwurf

Vorbereitung

- Arbeiten Sie die Ergebnisse aus dem Anforderungsreview in Ihre Use Cases ein.
- Analysieren Sie Ihre Use Cases und leiten Sie daraus Analyseklassen ab (noch keine Attribute oder Methoden). Zeichnen Sie ein entsprechendes Klassendiagramm auf einem Blatt Papier.
- Frischen Sie Ihr Wissen auf, über UML Klassendiagramme und das Suchen und Finden von Klassen (siehe auch Kapitel "Gutes Design" von OOAD) und über UML-Klassendiagramme
- Analysieren Sie das zugelieferte Rezeptbuch bis Sie die Funktionsweise verstehen
- Lesen Sie in der Anleitung zu MagicDraw den Abschnitt zu Klassendiagrammen. Schauen Sie sich genau an, wie man Datentypen anlegt (Type, Type Modifier und Container).

² Sie sind in der Lage die Begriffe zu erklären, wenn der Betreuer Sie dazu befragt.

³ Die Lieferung enthält eine h-Datei, .o-Datei, csv-Datei und main.cpp als Anwendungsbeispiel. Die csv-Datei enthält die Rezepte und muss im Ausführungsverzeichnis liegen. Wird die csv-Datei nicht gefunden, enthält das Rezeptbuch nur 5 Dummy-Rezepte.

2.1.) Klassendiagramm

Entwerfen Sie die erste Version des Klassenmodells indem Sie Ihre Use Cases analysieren und daraus Klassen ableiten. Legen Sie zuerst nur die Klassen an und wählen Sie aussagekräftige Namen (ohne Sonderzeichen und Umlaute). Anschließend dokumentieren Sie die Verantwortlichkeiten der Klassen⁴. Wenn die Klassenaufteilung stabil ist, ergänzen Sie Attribute (aber noch keine Operationen), Assoziation und Aggregationen/ Kompositionen. Prüfen Sie, ob es Generalisierungshierarchien gibt (welche Klassen haben gleiches Verhalten?). Beachten Sie die Regeln für einen guten Entwurf. Achten Sie dabei auf starken Zusammenhalt jeder Klasse und lose Kopplung der Klassen untereinander.

2.2.) Design eines "Mischbaren Rezeptbuchs"

In Ihrem Klassendiagramm taucht das Rezeptbuch auf. Eventuell haben Sie sogar das folgende "Problem" bereits festgestellt:

Das zugelieferte Rezeptbuch enthält viele Rezepte, von denen Sie aber nur wenige mischen können – je nachdem welche Zutaten zur Verfügung stehen. Überlegen Sie sich, welche Teile des Systems überhaupt von der Existenz dieser nicht-mischbaren Rezepte wissen müssen und kapseln Sie dieses Wissen in einem "Mischbaren Rezeptbuch", das nur noch die (mit den aktuellen Zutaten) mischbaren Rezepte anbietet.

2.3.) Verfeinerung des Designs

Erarbeiten Sie zu einigen Klassen (jedenfalls für das mischbare Rezeptbuch) die erforderlichen Methoden. Überlegen Sie sich, welche Klasse(n) in Ihrem System die mischbaren Rezepte bestimmen und welche Methoden Sie dafür einsetzen werden. Verfeinern Sie den Grobentwurf dieses Systemteils bis zu einem implementierbaren Design. Die Schnittstelle des "Mischbaren Rezeptbuchs" kann genau der Schnittstelle des zugelieferten Rezeptbuchs entsprechen. Erzeugen Sie bei Bedarf Konstruktoren, Destruktoren, Getter/Setter und verwenden Sie auch private-Methoden.

Ergebnis:

Ein Klassendiagramm beschreibt grob die Aufteilung des Systems in Klassen und Verantwortlichkeiten. Das Modell enthält Klassen mit ersten Methoden.

3. Termin: Design

Vorbereitung

- Frischen Sie Ihr Wissen auf, über UML-Sequenzdiagramme und Roundtrip-Engineering.
- Lesen Sie in der Anleitung zu MagicDraw die Abschnitte zu Sequenzdiagrammen und Code Generierung.
- Erarbeiten Sie ein Sequenzdiagramm das zeigt, wie die Initialisierung des Systems erfolgt⁵.

3.1.) Erster Implementierungszyklus

Das "mischbare Rezeptbuch" soll als erster Implementierungszyklus entworfen und umgesetzt werden. Der Betreuer hilft Ihnen bei der Erzeugung der Coderahmen. Dabei achten Sie auf Meldungen, die auf Zyklen hinweisen. Falls eine solche Meldung erscheint, rufen Sie unbedingt einen Betreuer. Kompilieren Sie die leeren Coderahmen. Wandeln Sie die zugelieferte main.cpp so ab, dass Ihre neue Klasse verwendet wird.

3.2.) Verfeinerung des Entwurfs

Verfeinern Sie Ihre Klassen so, dass Sie die Methoden zur Umsetzung der Anwendungsfälle beinhalten. Benennen Sie die Objekte sinnvoll (z.B. mit vorangestellten der/die/das oder ein/eine). Tipp: Investieren Sie besser hier etwas mehr Arbeit, bis Ihnen die Abläufe klar sind. Den Aufwand sparen Sie bei der Implementierung ein! Sie dürfen die Abläufe gerne als Sequenzdiagramme dokumentieren.

⁴ Verwenden Sie in MagicDraw die Eigenschaft "Documentation" für die jeweiligen Klassen.

⁵ Beachten Sie, dass Konstruktor-Aufrufe in Sequenzdiagrammen durch Pfeile auf das Objekt dargestellt werden.

3.3.) Erste Implementierung

Entwickeln Sie bis zum nächsten Termin den Code für das "Mischbare Rezeptbuch"⁶. Dabei wird die Zutatenliste eingelesen und die mischbaren Rezepte werden bestimmt.

Ergebnis

Der Entwurf ist soweit verfeinert, dass die Methoden zur Umsetzung definiert sind. Die Coderaahmen für das "Mischbare Rezeptbuch" sind generiert und kompilierbar.

Ihnen ist klar, wie die Umsetzung der Funktionalität mit den geforderten Methoden funktioniert.

4. Termin: Vorbereitung der restlichen Implementierung

Vorbereitung

- Implementieren Sie das "Mischbare Rezeptbuch" und testen Sie dessen Funktion inklusive des Einlesens der (zugefertigten) Zutatenliste. Ihr Programm zeigt an, welche der Rezepte mit den gegebenen Zutaten mischbar sind. Überprüfen Sie Ihr Ergebnis, indem Sie die Rezepte und die Zutaten (beides Textdateien) analysieren.
- Frischen Sie Ihr Wissen über das Beobachter-Muster auf und überlegen Sie sich mit einem Sequenzdiagramm wie Sie das Beobachter-Muster anwenden müssen.

4.1.) Reverse-Engineering

Zuerst muss der Code, den Sie für das mischbare Rezeptbuch implementiert haben, in das Modell importiert werden. Dazu führen Sie ein Reverse-Engineering dieser Klassen durch. Der Betreuer hilft Ihnen dabei.

4.2.) Beobachter-Muster

Identifizieren Sie die zur Anwendung des Beobachter-Musters notwendigen Klassen. Falls noch Klassen fehlen, legen Sie diese bitte (ohne jegliche Details) an. Anschließend werden die Klassen durch MagicDraw mit den entsprechenden Methoden versehen.

Legen Sie ein neues Klassendiagramm an und stellen Sie die Klassen der Observer-Musters so dar, dass das Beobachtermuster als solches erkennbar ist

Zeichnen Sie unter Berücksichtigung des Beobachter-Muster 1-2 Sequenzdiagramme, die zeigen, wie beim Dosieren die Interaktion zwischen einem Dosierer, dem Display und der Waage funktioniert.

Zeigen Sie, wie bei der Dosierung die Gewichtserhöhung bei der Waage simuliert wird.

4.3.) Aufsetzen des Roundtrip-Engineering

Bevor Sie diesen Punkt angehen, sollte Ihr Modell konsistent und vollständig sein. Prüfen Sie außerdem in der "Dependency Matrix", ob Ihr Entwurf Zyklen enthält (Gegebenenfalls rufen Sie einen Betreuer, der Ihnen hilft den Zyklus aufzulösen).

Warnung: Falls Sie mit einem unvollständigen oder fehlerhaften Modell weitermachen, wird Sie das später erheblichen Zusatzaufwand kosten!

Erzeugen Sie durch Forward Engineering die Coderaahmen. Dabei achten Sie erneut auf Meldungen, die auf Zyklen hinweisen.

Kompilieren Sie alle Coderaahmen zusammen mit einer „Hello World“-Main-Funktion und linken Sie das zugefertigte Rezeptbuch dazu (siehe auch Anhang A). Beginnen Sie noch nicht mit der eigentlichen Implementierung, sondern machen Sie lediglich erforderliche Anpassungen, bis der Code kompiliert. Importieren Sie anschließend den Code durch Reverse-Engineering, erzeugen Sie den Code neu und überprüfen Sie noch einmal die Kompilierbarkeit.

4.4.) Implementierung

Entwickeln Sie **bis zum Beginn (!!!) des nächsten Termins** den Rest des Systems. Halten Sie sich bei der Implementierung an die Entwurfsentscheidungen in Ihren Klassen- und Sequenzdiagrammen.

⁶ Bei seltsamen Effekten mit Stringvergleichen beachten Sie bitte den Hinweis zu den unterschiedlichen Dateiformaten von Windows und Unix.

Tipps:

- Lesen Sie die Hinweise auf Seite 6.
- Nutzen Sie die Zerlegung Ihrer Systems in Klassen zur Aufteilung der Implementierung innerhalb Ihres Teams.
- Achten Sie bei der Implementierung darauf, dass der Code und das Modell konsistent bleiben. Änderungen an der Signatur von Klassen/Methoden nehmen Sie am besten nur im Code vor. Das Ergebnis können Sie dann im nächsten Praktikum in Ihr Modell importieren (Reverse Engineering).
- Bei größeren Problemen bei der Implementierung ist oft ein falsches Design die Ursache. Es ist dann geschickter, einen Schritt zurück zu gehen und das Design zu überarbeiten. Falls Sie schon Code geschrieben haben, müssen Sie aber zuerst den Code importieren bevor Sie Änderungen im Modell machen können.

Ergebnis

Ein vollständiges Design, das die zu implementierenden Klassen und Methoden beschreibt und mit Hilfe von Sequenzdiagrammen "getestet" wurde und. Leere Coderahmen, die aber schon kompilierbar sind.

5. Termin: Abgabe des Produkts beim Kunden und Dokumentation

Dieser Termin entspricht dem Abgabetermin bei Ihrem Auftraggeber.

Vorbereitung

Zu **Beginn** des Termins muss Ihr Code bereits vollständig und lauffähig sein.

Achtung! Sollte Ihr Code unvollständig oder offensichtlich nicht lauffähig sein, so erhalten Sie sofort eine rote Karte und sind durchgefallen. Das ist sehr streng? Dann überlegen Sie einmal was Ihr Auftraggeber mit Ihnen machen würde, wenn Sie am Abgabetermin ohne brauchbares Ergebnis auftauchen...

Sie sollten Sie also lieber nicht bis zum letzten Tag mit der Lösung der Aufgabe warten.

5.1.) Importieren des Codes

Mit Reverse-Engineering werden Ihr Code und eventuelle Änderungen an Klassen oder Methoden in Ihr Modell übernommen. Fehler in diesem Arbeitsschritt können leicht Ihr Modell zerstören. Fragen Sie deshalb im Zweifelsfall Ihren Betreuer anstatt herum zu probieren.

5.2.) Zusammenstellung der Dokumentation

Erzeugen Sie eine Dokumentation mit MagicDraw und speichern Sie das Ergebnis als PDF-Datei.

5.3.) Endabnahme durch den Anwender

Die Endabnahme wird durch den Auftraggeber durchgeführt. Zur Abnahme gehört auch das Prüfen des Modells mit Dokumentation. Prüfen Sie vorher selbst, ob Ihr System die Abnahmetests, die den Use Cases aus der ersten Übung entsprechen, besteht.

Ergebnis

Ein wartbares System, das die Anforderungen erfüllt und den Abnahmetest bestanden hat.

6. Termin: Qualitätssicherung

Das Ziel des 6. Praktikums ist die Analyse Ihrer in C++ geschriebenen Anwendung im Hinblick auf Qualität. Dazu werden potenzielle Änderungsanforderungen diskutiert und anschließend der Code mit dem Test- und Analysetool Logiscope der Firma Telelogic untersucht.

Vorbereitung

Drucken Sie die Aufgaben im Anhang B aus damit Sie die Antworten eintragen können.

Frischen Sie Ihr Wissen auf, über Metriken und deren Bedeutung.

Falls Sie noch andere C++ - Projekte haben und an Feedback zur Qualität der Implementierung interessiert sind, spielen Sie diesen Code bitte vorher in Ihren Bereich auf dem UServ ein⁷. Im Praktikum können Sie auch diesen Code analysieren.

⁷ Falls Sie keine eigenen Projekte haben, können Sie natürlich auch Open Source Software analysieren.

6.1.) Diskussion Änderbarkeit

In der Gruppe werden folgende Änderungen und die Auswirkungen auf Ihr Modell diskutiert

- es wird ein Rezeptbuch mit 1000 neuen Rezepten zugeliefert
- das User-Interface soll in Deutsch und Englisch verfügbar sein
- es wird in den Rezepten ein neuer Verarbeitungsschritt "Schütteln" eingeführt
- es soll bei einem Füllstand von 200g ein Warnungs-Piepton ausgegeben werden
- die gleiche Zutat kann in mehreren Dosierern sein (es soll automatisch umgeschaltet werden, wenn ein Dosierer leer ist)
- die Cocktails sollen schneller gemischt werden, indem die Zutaten parallel abgewogen werden. Dazu erhält jeder Dosierer eine eigene Messvorrichtung.

Bitte analysieren Sie Ihr Modell für die obigen Änderungen mit folgenden Fragen:

- In welchen Klassen und Methoden müssen Sie Änderungen machen?
- Ist die Änderung eher ein Hack oder passt sie ins Design?
- Könnte die Änderungen auch leicht jemand anderes machen?
- Wie viele Stunden brauchen Sie zur Umsetzung?

6.2.) Metriken

Die Erläuterungen zum Tool und die Aufgaben finden Sie im Anhang B ab Seite 13.

Ergebnis

Sie haben die Grenzen Ihres Entwurfs erkannt und den Vorteil von Metriken verstanden. Sie kennen die "Problemstellen" in Ihrem Code. Sie wissen jetzt, warum diese Stellen problematisch sind, und wie man die Aufgabe geschickter lösen könnte.

Sie haben den Vorteil eines Modells für die Wartbarkeit und Änderbarkeit verstanden.

Anhang A

Verwendung des Linkers

Nach dem Kompilieren des Sourcecodes wird der Linker aufgerufen, der die vom Compiler erzeugten Binärdateien zusammenfügt. Das heißt, die Aufrufe von Methoden und Klassen werden mit deren Binärcode verknüpft. Beim "normalen" Kompilieren von C++-Dateien erfolgt der Linkeraufruf automatisch und Sie bemerken den Linker kaum (bis auf gelegentliche unverständliche Fehlermeldungen). Für das zugelierte Rezeptbuch müssen Sie allerdings die entsprechende Binärdatei manuell zum Linkvorgang hinzufügen. Wie das für die gängigsten Compiler/IDEs geht, steht hier:

Das Rezeptbuch wird (auf der Homepage) zur Verfügung gestellt

- als Binärdatei (.o, .obj) für verschiedene OSs, CPUs und Compiler
Sie müssen natürlich die Datei verwenden, die zu Ihrem Rechner passt – also unterschiedliche Dateien auf Linux, PC, MAC – und dann auch noch abhängig von Ihrem Compiler.
Verwenden Sie die falsche Datei, gibt es üble Fehlermeldungen vom Linker.
- mit einer Header-Datei (.h) und
- einem Anwendungsbeispiel (main.cpp)
- Der Source-Code ist nicht zugänglich

a) GNU-Compiler (g++)

- Aufruf: `g++ main.cpp -o Rezeptbuch.exe RezeptbuchXXX.o`

b) Microsoft Visual Studio 2010 & Co.:

- Einfach das passende Binary (Rezeptbuch....obj) mit Drag & Drop in den Ordner "Ressourcendateien" in die Projektmappe werfen.

c) Netbeans:

- Wählen Sie mit der rechten Maustaste Resource Files -> Add Existing Item und wählen Sie das Binary aus.
- Nun müssen die Kompiliereinstellung geändert werden. Hierzu wieder ein Rechtsklick auf das eingefügte Objekt und Eigenschaften wählen. Hier kann nun unter Tool "C++ Compiler" ausgewählt werden.
- Zu guter Letzt muss die Objektdaten noch in den Ordner build/Debug/GNU-Linux-x86/ kopiert werden.

Anhang B

6. Praktikum - Softwaremetriken

Im 6. Praktikum analysieren Sie Ihre in C++ geschriebenen Anwendung mit dem Test- und Analysetool Logiscope der Firma Telelogic. Mit Logiscope kann Software automatisch vermessen und analysiert werden. Die Qualitätsprüfung basiert auf einem Qualitätsmodell, das objektive Maße für Software definiert. Diese objektiven Maße (Metrics) werden am Quellcode gemessen. Daraus werden über mathematische Gleichungen Qualitätseigenschaften (Factors, Criteria) berechnet. Der Wertebereich für die Qualitätseigenschaften wird in Qualitätsstufen (Excellent, Good, Fair, Poor) eingeteilt über die die erreichte Qualität für eine Anwendung oder Teile einer Anwendung berechnet werden kann.

Im Praktikum werden folgende Logiscope-Komponenten eingesetzt.

Logiscope Studio

Das Studio-Paket wird verwendet, um ein Projekt zu definieren. Es misst die Metrics für eine Anwendung und berechnet daraus die Factors und Criteria. Das gesamte Qualitätsmodell ist erreichbar und kann an die Anforderungen eines Projektes angepasst werden. Alle gemessenen Werte für Funktionen, Klassen und die Gesamtanwendung können angesehen werden. Der Qualitätsreport für die Gesamtanwendung, Klassen und Funktionen kann geöffnet werden.

Logiscope Viewer

Mit dem Logiscope Viewer können die Messwerte in verschiedenen Diagrammen grafisch dargestellt werden.

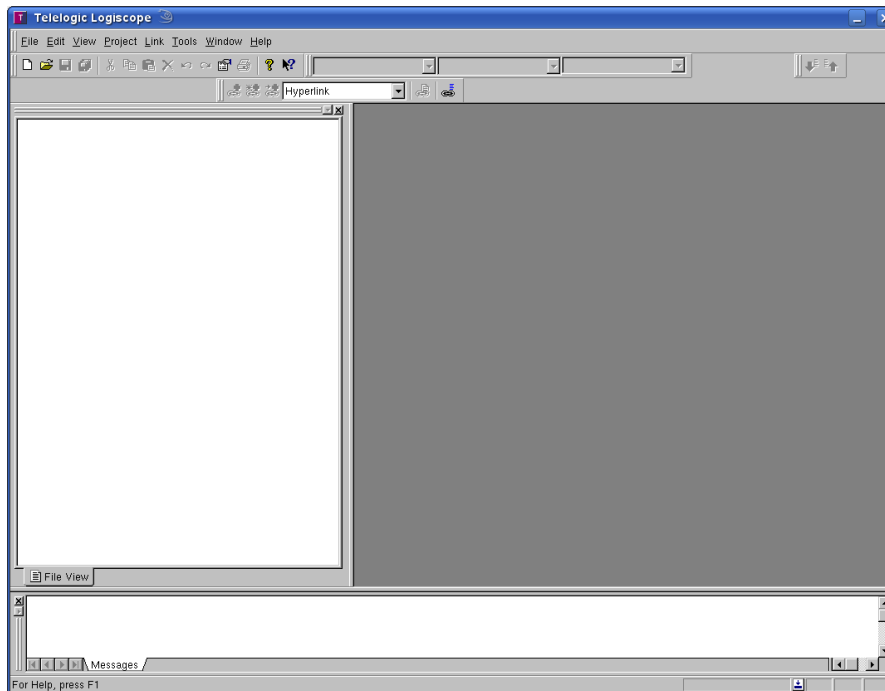
Praktikumsdurchführung

Grundlage für die Qualitätssicherung ist ihr Projekt mit den C++-Sourcen.

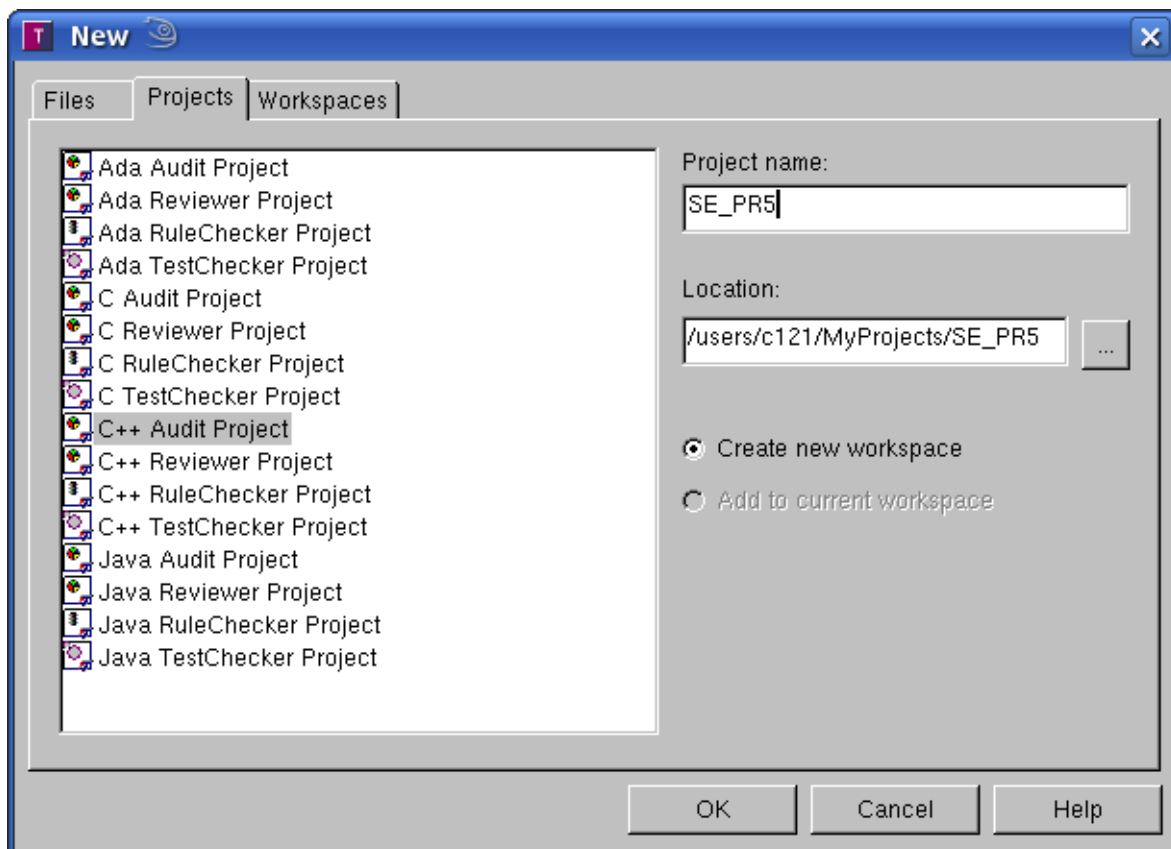
Start von Logiscope

Der Start des Programms wird Ihnen beim ersten Mal von Ihrem Betreuer erklärt. Starten Sie anschließend das Tool **Logiscope** durch Eingabe von **vcs&** in einer Shell.

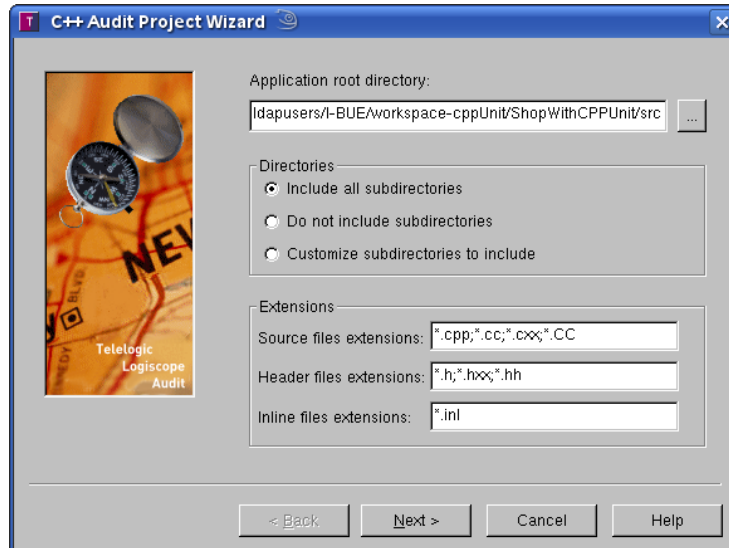
Hinweis: Bitte warten Sie einen Moment bis die Anwendung gestartet ist.



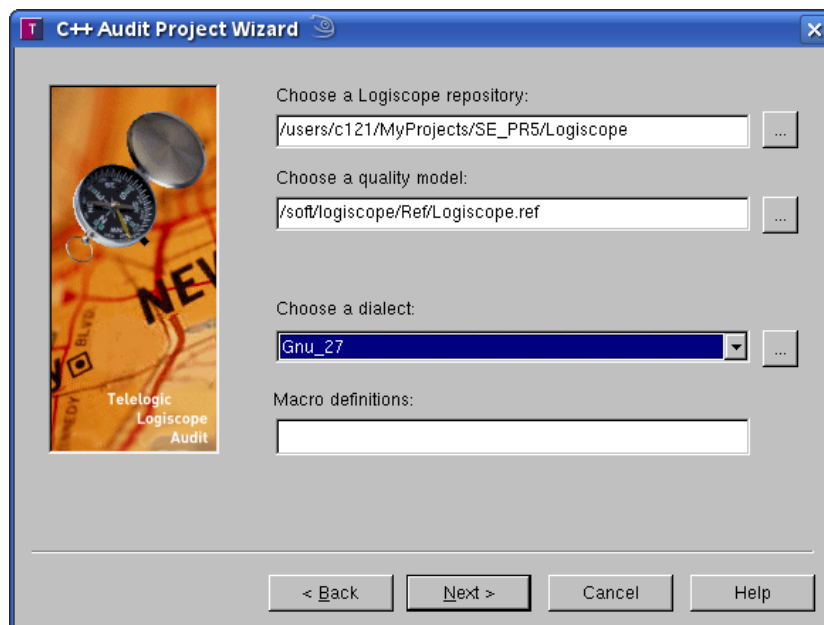
Erzeugen Sie über **File/New** ein neues Projekt. Wählen Sie **C++Audit Project** als Projekt-Typ. Geben Sie einen Namen für das Projekt an, z. B. SE_PR5, und bestätigen Sie mit OK. Daraufhin wird ein neuer Arbeitsbereich für ihr Projekt angelegt.



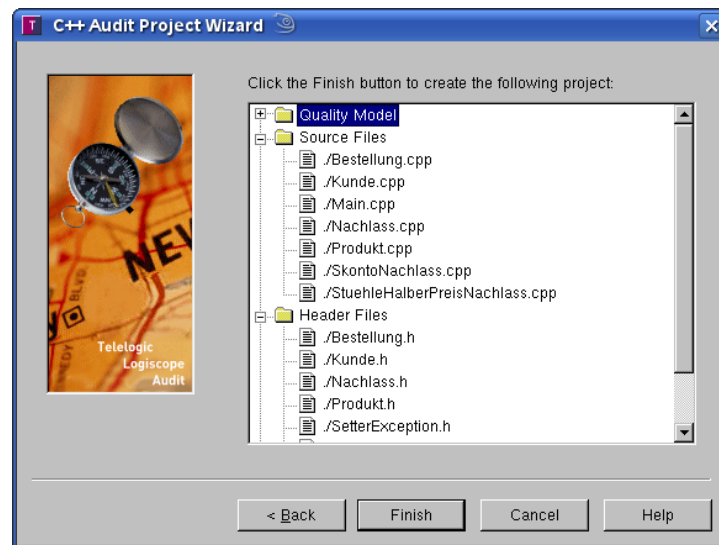
Im folgenden Fenster stellen Sie das *Application root directory* ein. Das ist das Verzeichnis, das den zu prüfenden Code enthält. Geben Sie hier ihr Source-Verzeichnis an. Falls Ihr Verzeichnis weitere Ordner enthält (z.B. mit Sicherungen des Quellcodes), dann wählen Sie "do not include subdirectories". Anschließend klicken Sie auf „Next“.



Wählen Sie als Parser-Konfiguration **Gnu_27** und bestätigen die Auswahl mit *Next*. Das Qualitätsmodell und das Logiscope Repository sind schon korrekt eingestellt und müssen nicht verändert werden.

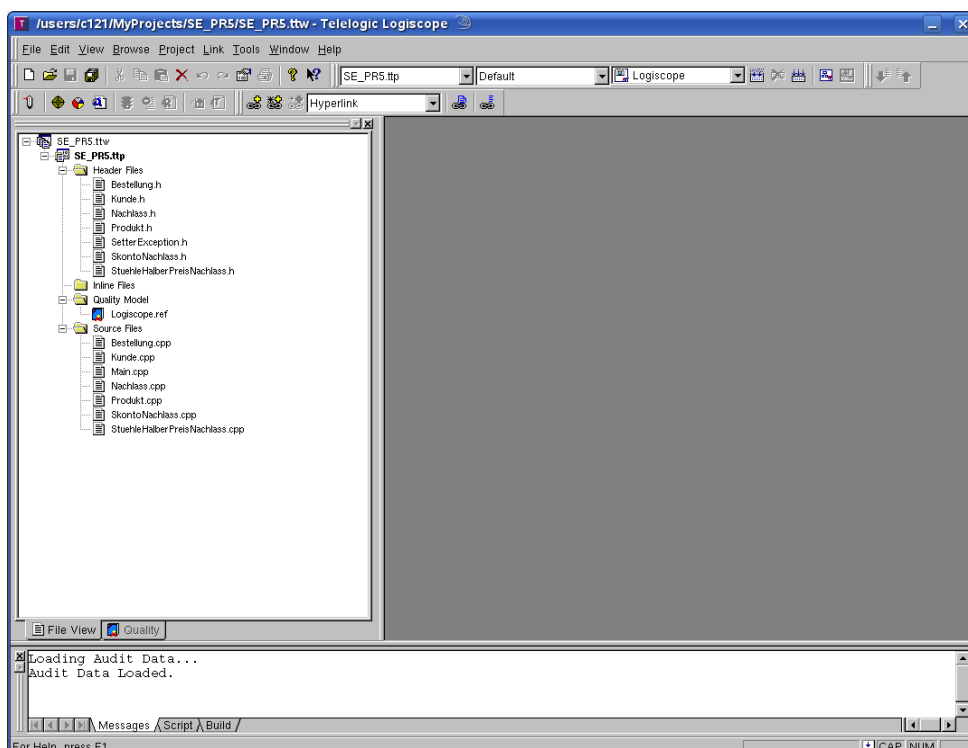
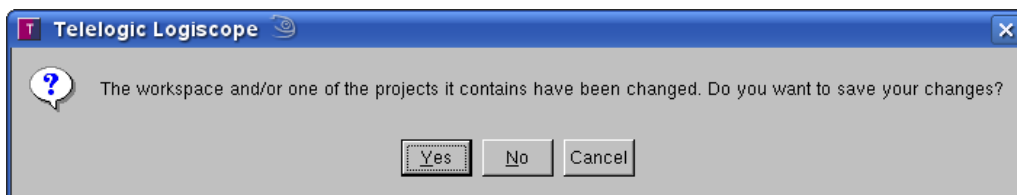


Überprüfen Sie im folgenden Fenster die vorgenommenen Einstellungen. Unter *Source Files* müssen alle zu überprüfenden cpp-Dateien enthalten sein. Unter *Header Files* stehen alle h-Dateien. Wenn die Einstellungen stimmen, klicken Sie auf Finish. Andernfalls gehen Sie über *Back* zurück, um Ihre vorherigen Einstellungen zu korrigieren.

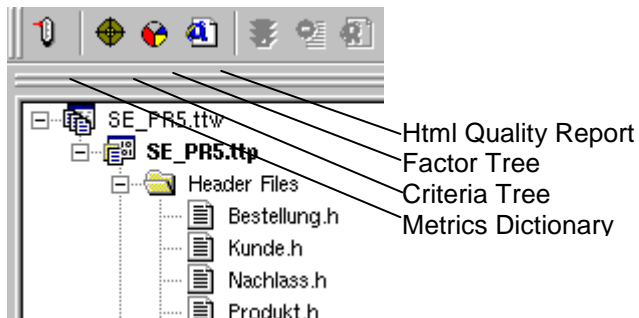


Arbeiten mit Logiscope

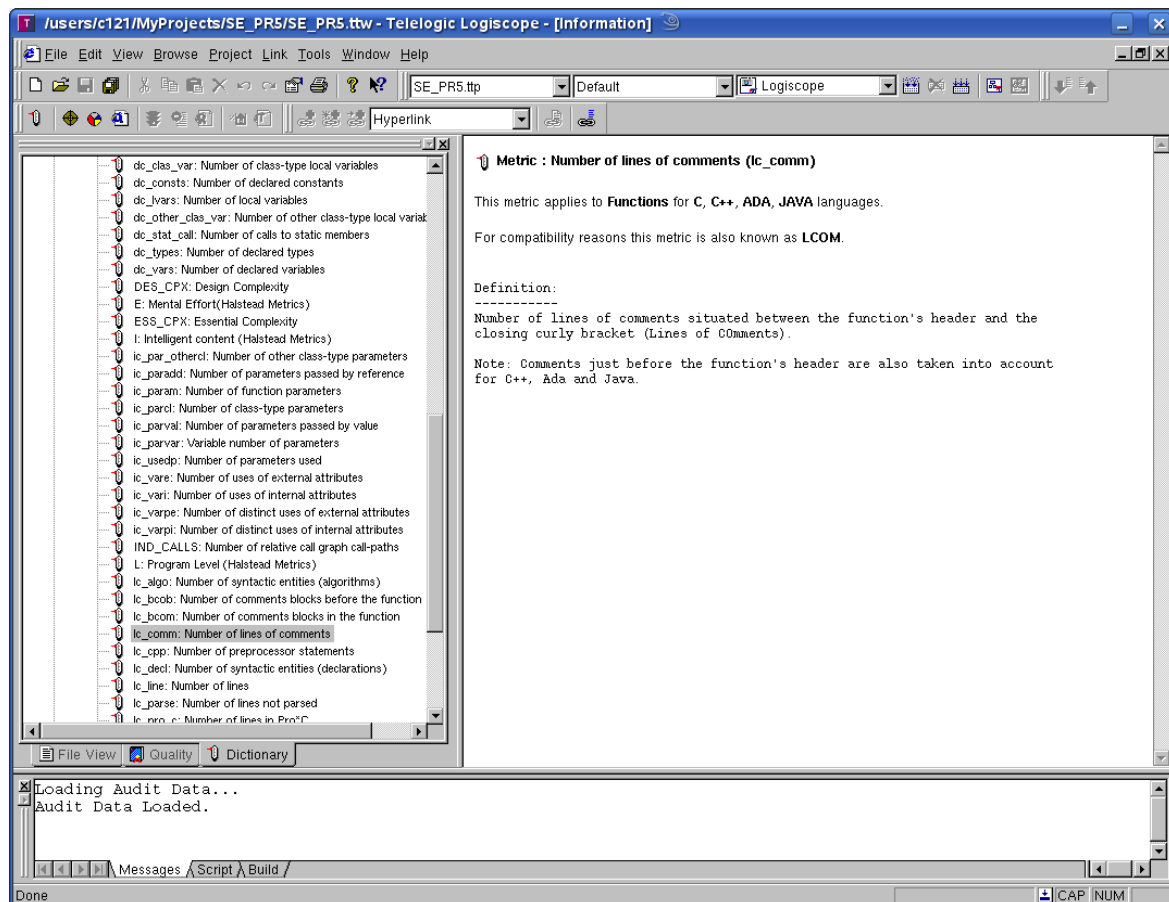
Nachdem nun alle Projekteinstellungen korrekt sind, können wir mit dem Werkzeug unsere Anwendung untersuchen. Starten Sie zunächst die *Qualitätsüberprüfung* für den zu untersuchenden Code über das Menu **Project/Build**. Sofern ein Bestätigungsfenster erscheint, klicken Sie auf „Yes“.



Im linken Frame des Logiscope Studio können Sie sich wahlweise die zu untersuchenden Dateien oder das Qualitätsmodell, gegen das geprüft wird, anzeigen lassen. Über die Toolbar können folgende Funktionen erreicht werden:



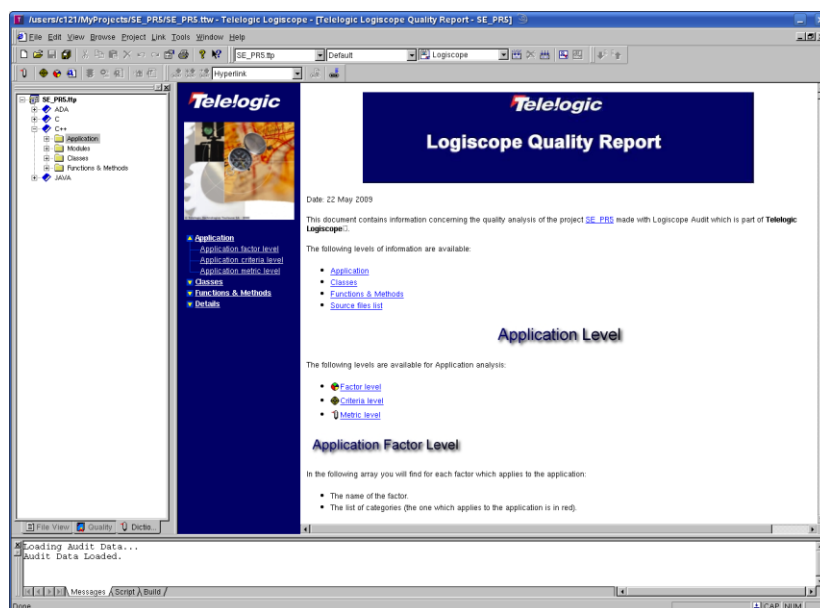
Öffnen Sie das „Metrics Dictionary“ und suchen Sie Erklärungen zu nachfolgenden Metriken (ergänzen Sie die rechte Tabellenspalte):



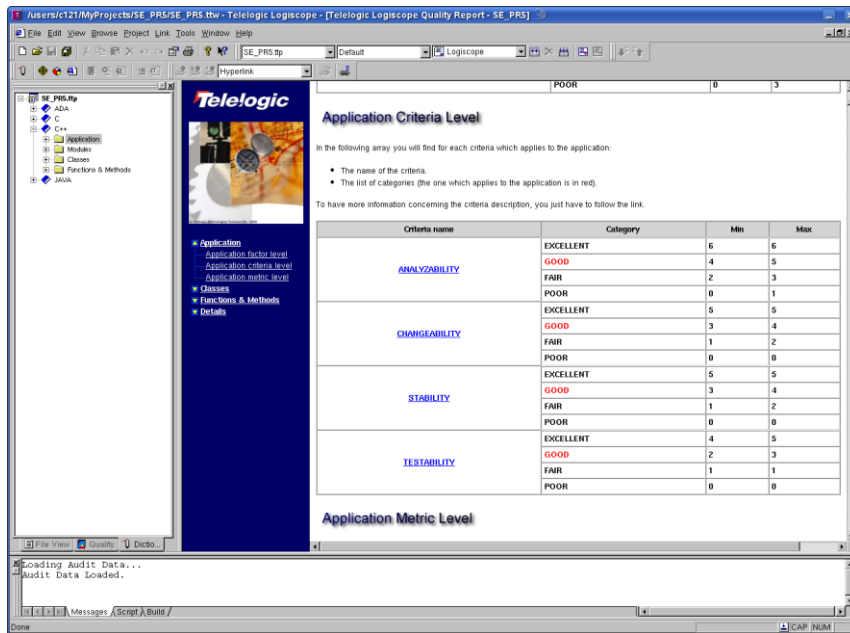
Metric	Erklärung
lc_comm	
lc_stat	
ct_vg	
ct_bran	
ct_path	
dc_calling	
dc_calls	

Erkennen Sie einen Zusammenhang zwischen diesen Metriken und der Qualität von Code?

Als nächstes erzeugen Sie den *Html Quality Report* und schauen sich die Ergebnisse an. Der Bericht ist unterteilt in verschiedene Untersuchungsebenen (Levels): Application, Classes, Function & Methods und Source Files.



Schauen Sie sich die Ergebnisse der Bewertung an für die Anwendungs-Kriterien Analysierbarkeit, Änderbarkeit, Stabilität und Testbarkeit.



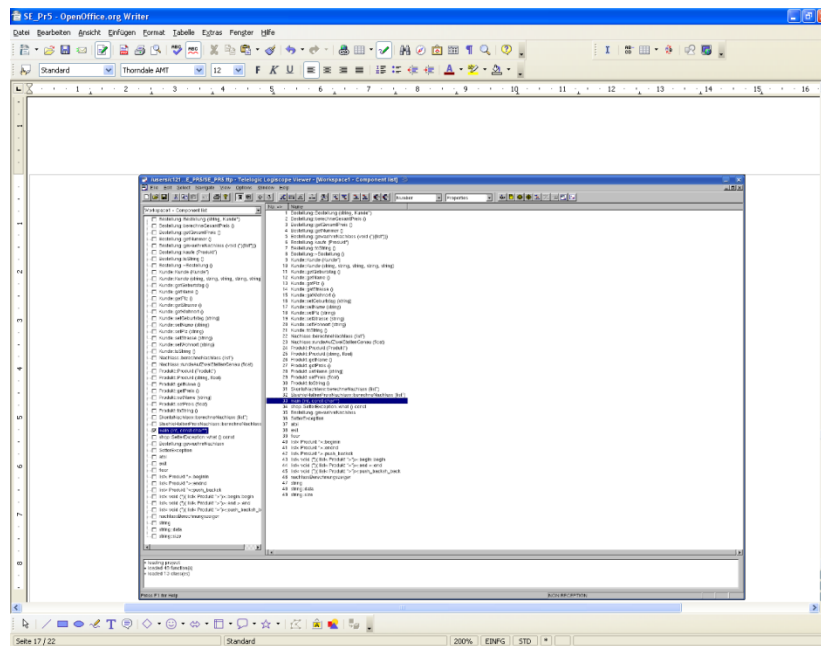
Notieren Sie in der nachfolgenden Tabelle, wie diese Kriterien aus den Basis-Metriken abgeleitet werden.

Metrik	Erklärung
Analysierbarkeit	
Änderbarkeit	
Stabilität	
Testbarkeit	

Stimmen Sie diesen Kriterien zu?

Grafische Darstellungen

Über das Menü **Project/Start Viewer** können Sie den Logiscope Viewer starten und den Qualitätsreport in verschiedenen Grafiken betrachten.



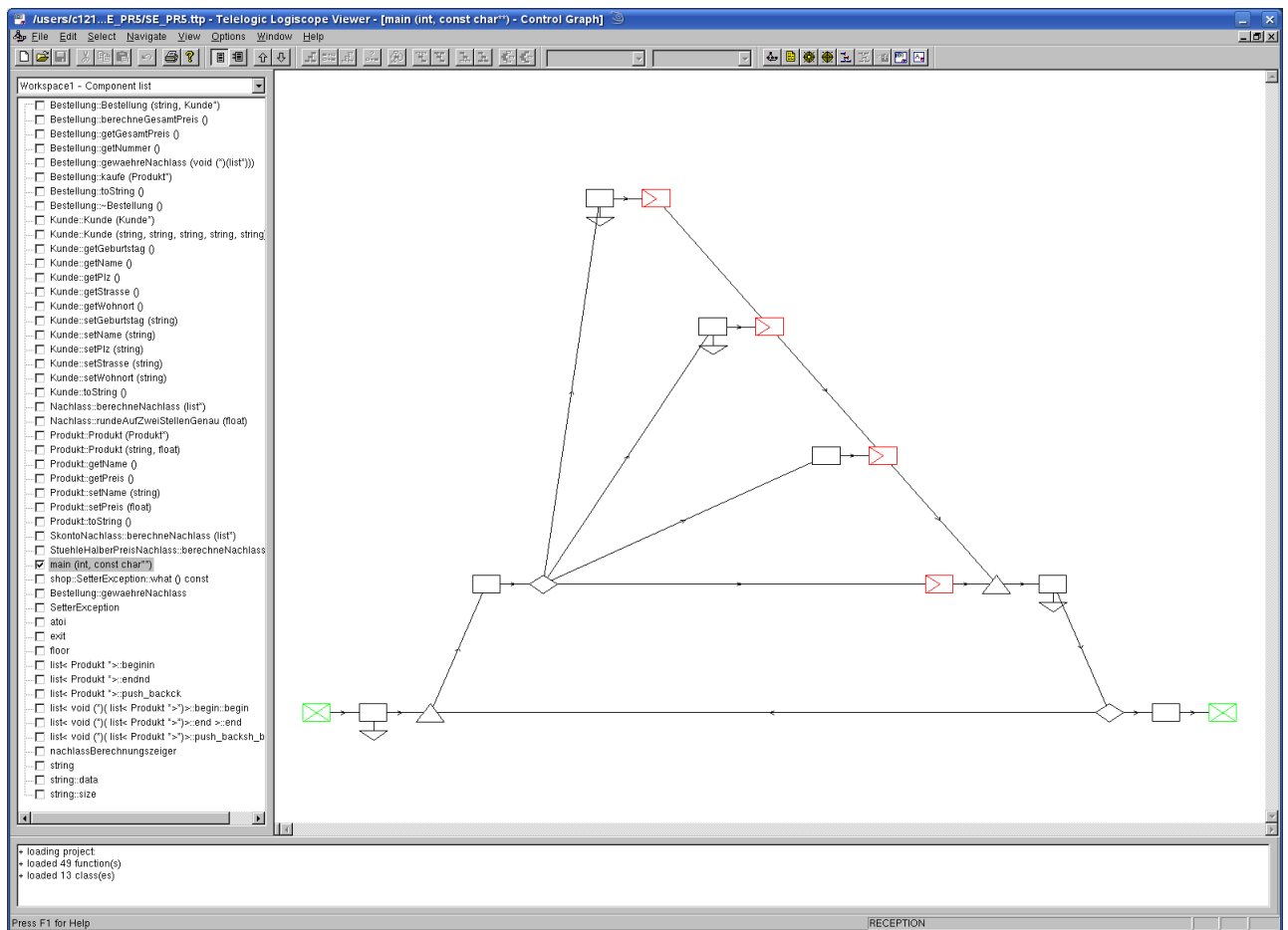
Sie können die darzustellenden Diagramme über die *Control Center Toolbar* auswählen.



Die Buttons haben (von links nach rechts) folgende Bedeutung:

- Anzeige des Kontrollflussgraphen für eine Funktion.
- Anzeige des Quellcodes für eine Funktion. Die aktuelle Zeile ist markiert.
- Anzeige des Metrik-Kiviatgraphen für die selektierte Funktion.
- Anzeige des Criteria- Kiviatgraphen für die selektierte Funktion.
- Anzeige des (relativen) Aufrufgraphen für die Funktion bzw. Anwendung.

Selektieren Sie Ihre main-Methode und lassen Sie sich den Kontrollflussgraphen anzeigen.

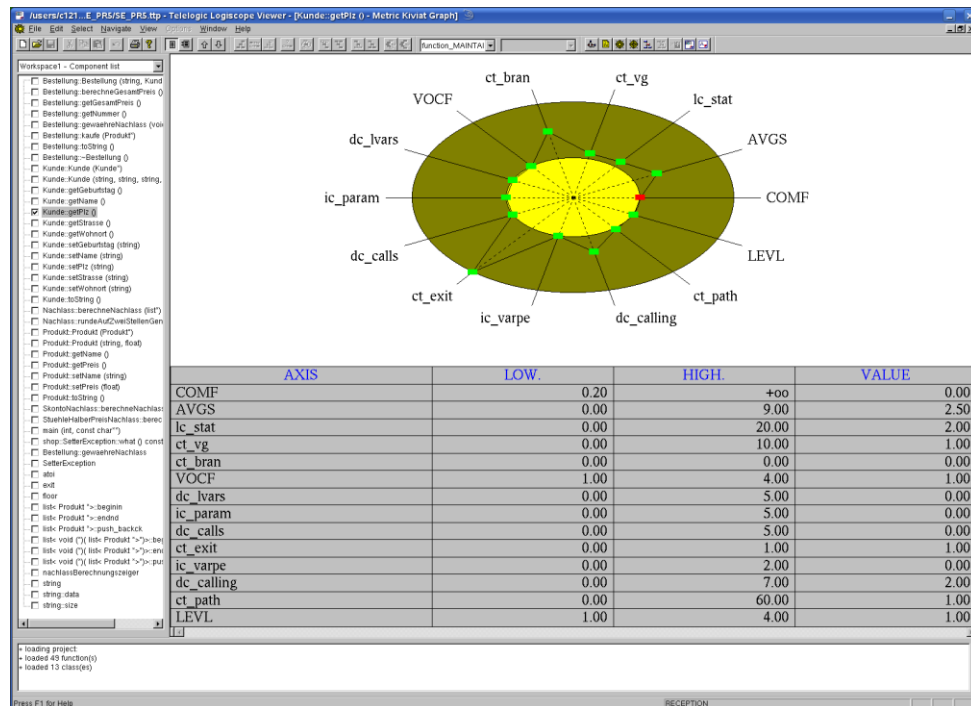


Berechnen Sie (manuell) den Wert für die zyklomatische Komplexität der Methode **main()**.

Antwort: $z(G) =$ _____

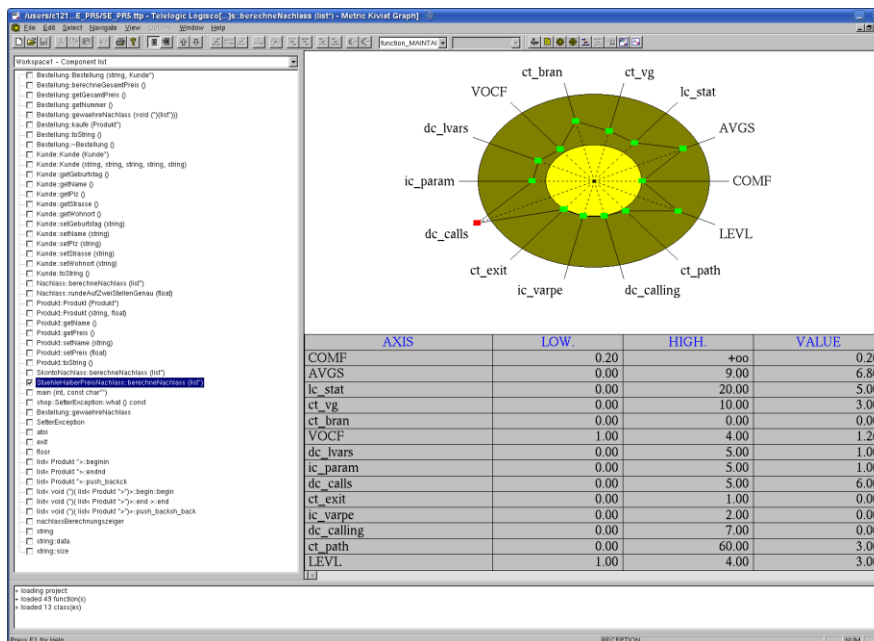


Selektieren Sie die Methode in der Sie die Cocktails mischen und öffnen Sie den Metric Kiviat Graph.



Falls der Wert der Metrik **COMF** zu niedrig ist, ändern Sie die Methode im Source-Code so ab, damit der Wert auf mindestens 1 liegt. *Hinweis:* nach jeder Software-Änderung muss ein Rebuild des Logiscope-Projektes (Project\Rebuild all) erfolgen.

Selektieren Sie die Methode in der Sie die mischbaren Rezepte herausfiltern und öffnen Sie den Metric Kiviat Graph.



```

1  /*
2  * Hochschule Darmstadt, FB Informatik
3  * Praktikum : Software Engineering - Prof. Dr. Frank Böhler
4  *
5  * Datei : Main.cpp
6  *
7  * Erstellt am: 10. März 2009
8  * Autor : jo
9  *
10 *
11 */
12
13 #include "StuehleHalberPreisNachlass.h"
14
15 using shop::StuehleHalberPreisNachlass;
16
17 void StuehleHalberPreisNachlass::berechneNachlass(list<Produkt*>* bestellteProdukte)
18 {
19     // Verringere den Preis jedes Produktes, dass den Namen "Stuhl" trägt auf die Hälfte
20     for(list<Produkt*>::iterator iter = bestellteProdukte->begin(); iter != bestellteProdukte->end(); iter++)
21     {
22         if((*iter)->getName() == "Stuhl")
23             (*iter)->setPreis( rundeAufZweiStellenGenau( (*iter)->getPreis() * 0.5 ) );
24     }
25 }
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999
1000
1001
1002
1003
1004
1005
1006
1007
1008
1009
1010
1011
1012
1013
1014
1015
1016
1017
1018
1019
1020
1021
1022
1023
1024
1025
1026
1027
1028
1029
1030
1031
1032
1033
1034
1035
1036
1037
1038
1039
1040
1041
1042
1043
1044
1045
1046
1047
1048
1049
1050
1051
1052
1053
1054
1055
1056
1057
1058
1059
1060
1061
1062
1063
1064
1065
1066
1067
1068
1069
1070
1071
1072
1073
1074
1075
1076
1077
1078
1079
1080
1081
1082
1083
1084
1085
1086
1087
1088
1089
1090
1091
1092
1093
1094
1095
1096
1097
1098
1099
1100
1101
1102
1103
1104
1105
1106
1107
1108
1109
1110
1111
1112
1113
1114
1115
1116
1117
1118
1119
1120
1121
1122
1123
1124
1125
1126
1127
1128
1129
1130
1131
1132
1133
1134
1135
1136
1137
1138
1139
1140
1141
1142
1143
1144
1145
1146
1147
1148
1149
1150
1151
1152
1153
1154
1155
1156
1157
1158
1159
1160
1161
1162
1163
1164
1165
1166
1167
1168
1169
1170
1171
1172
1173
1174
1175
1176
1177
1178
1179
1180
1181
1182
1183
1184
1185
1186
1187
1188
1189
1190
1191
1192
1193
1194
1195
1196
1197
1198
1199
1200
1201
1202
1203
1204
1205
1206
1207
1208
1209
1210
1211
1212
1213
1214
1215
1216
1217
1218
1219
1220
1221
1222
1223
1224
1225
1226
1227
1228
1229
1230
1231
1232
1233
1234
1235
1236
1237
1238
1239
1240
1241
1242
1243
1244
1245
1246
1247
1248
1249
1250
1251
1252
1253
1254
1255
1256
1257
1258
1259
1260
1261
1262
1263
1264
1265
1266
1267
1268
1269
1270
1271
1272
1273
1274
1275
1276
1277
1278
1279
1280
1281
1282
1283
1284
1285
1286
1287
1288
1289
1290
1291
1292
1293
1294
1295
1296
1297
1298
1299
1300
1301
1302
1303
1304
1305
1306
1307
1308
1309
1310
1311
1312
1313
1314
1315
1316
1317
1318
1319
1320
1321
1322
1323
1324
1325
1326
1327
1328
1329
1330
1331
1332
1333
1334
1335
1336
1337
1338
1339
1340
1341
1342
1343
1344
1345
1346
1347
1348
1349
1350
1351
1352
1353
1354
1355
1356
1357
1358
1359
1360
1361
1362
1363
1364
1365
1366
1367
1368
1369
1370
1371
1372
1373
1374
1375
1376
1377
1378
1379
1380
1381
1382
1383
1384
1385
1386
1387
1388
1389
1390
1391
1392
1393
1394
1395
1396
1397
1398
1399
1400
1401
1402
1403
1404
1405
1406
1407
1408
1409
1410
1411
1412
1413
1414
1415
1416
1417
1418
1419
1420
1421
1422
1423
1424
1425
1426
1427
1428
1429
1430
1431
1432
1433
1434
1435
1436
1437
1438
1439
1440
1441
1442
1443
1444
1445
1446
1447
1448
1449
1450
1451
1452
1453
1454
1455
1456
1457
1458
1459
1460
1461
1462
1463
1464
1465
1466
1467
1468
1469
1470
1471
1472
1473
1474
1475
1476
1477
1478
1479
1480
1481
1482
1483
1484
1485
1486
1487
1488
1489
1490
1491
1492
1493
1494
1495
1496
1497
1498
1499
1500
1501
1502
1503
1504
1505
1506
1507
1508
1509
1510
1511
1512
1513
1514
1515
1516
1517
1518
1519
1520
1521
1522
1523
1524
1525
1526
1527
1528
1529
1530
1531
1532
1533
1534
1535
1536
1537
1538
1539
1540
1541
1542
1543
1544
1545
1546
1547
1548
1549
1550
1551
1552
1553
1554
1555
1556
1557
1558
1559
1560
1561
1562
1563
1564
1565
1566
1567
1568
1569
1570
1571
1572
1573
1574
1575
1576
1577
1578
1579
1580
1581
1582
1583
1584
1585
1586
1587
1588
1589
1590
1591
1592
1593
1594
1595
1596
1597
1598
1599
1600
1601
1602
1603
1604
1605
1606
1607
1608
1609
1610
1611
1612
1613
1614
1615
1616
1617
1618
1619
1620
1621
1622
1623
1624
1625
1626
1627
1628
1629
1630
1631
1632
1633
1634
1635
1636
1637
1638
1639
1640
1641
1642
1643
1644
1645
1646
1647
1648
1649
1650
1651
1652
1653
1654
1655
1656
1657
1658
1659
1660
1661
1662
1663
1664
1665
1666
1667
1668
1669
1670
1671
1672
1673
1674
1675
1676
1677
1678
1679
1680
1681
1682
1683
1684
1685
1686
1687
1688
1689
1690
1691
1692
1693
1694
1695
1696
1697
1698
1699
1700
1701
1702
1703
1704
1705
1706
1707
1708
1709
1710
1711
1712
1713
1714
1715
1716
1717
1718
1719
1720
1721
1722
1723
1724
1725
1726
1727
1728
1729
1730
1731
1732
1733
1734
1735
1736
1737
1738
1739
1740
1741
1742
1743
1744
1745
1746
1747
1748
1749
1750
1751
1752
1753
1754
1755
1756
1757
1758
1759
1760
1761
1762
1763
1764
1765
1766
1767
1768
1769
1770
1771
1772
1773
1774
1775
1776
1777
1778
1779
1780
1781
1782
1783
1784
1785
1786
1787
1788
1789
1790
1791
1792
1793
1794
1795
1796
1797
1798
1799
1800
1801
1802
1803
1804
1805
1806
1807
1808
1809
1810
1811
1812
1813
1814
1815
1816
1817
1818
1819
1820
1821
1822
1823
1824
1825
1826
1827
1828
1829
1830
1831
1832
1833
1834
1835
1836
1837
1838
1839
1840
1841
1842
1843
1844
1845
1846
1847
1848
1849
1850
1851
1852
1853
1854
1855
1856
1857
1858
1859
1860
1861
1862
1863
1864
1865
1866
1867
1868
1869
1870
1871
1872
1873
1874
1875
1876
1877
1878
1879
1880
1881
1882
1883
1884
1885
1886
1887
1888
1889
1890
1891
1892
1893
1894
1895
1896
1897
1898
1899
1900
1901
1902
1903
1904
1905
1906
1907
1908
1909
1910
1911
1912
1913
1914
1915
1916
1917
1918
1919
1920
1921
1922
1923
1924
1925
1926
1927
1928
1929
1930
1931
1932
1933
1934
1935
1936
1937
1938
1939
1940
1941
1942
1943
1944
1945
1946
1947
1948
1949
1950
1951
1952
1953
1954
1955
1956
1957
1958
1959
1960
1961
1962
1963
1964
1965
1966
1967
1968
1969
1970
1971
1972
1973
1974
1975
1976
1977
1978
1979
1980
1981
1982
1983
1984
1985
1986
1987
1988
1989
1990
1991
1992
1993
1994
1995
1996
1997
1998
1999
2000
2001
2002
2003
2004
2005
2006
2007
2008
2009
2010
2011
2012
2013
2014
2015
2016
2017
2018
2019
2020
2021
2022
2023
2024
2025
2026
2027
2028
2029
2030
2031
2032
2033
2034
2035
2036
2037
2038
2039
2040
2041
2042
2043
2044
2045
2046
2047
2048
2049
2050
2051
2052
2053
2054
2055
2056
2057
2058
2059
2060
2061
2062
2063
2064
2065
2066
2067
2068
2069
2070
2071
2072
2073
2074
2075
2076
2077
2078
2079
2080
2081
2082
2083
2084
2085
2086
2087
2088
2089
2090
2091
2092
2093
2094
2095
2096
2097
2098
2099
2100
2101
2102
2103
2104
2105
2106
2107
2108
2109
2110
2111
2112
2113
2114
2115
2116
2117
2118
2119
2120
2121
2122
2123
2124
2125
2126
2127
2128
2129
2130
2131
2132
2133
2134
2135
2136
2137
2138
2139
2140
2141
2142
2143
2144
2145
2146
2147
2148
2149
2150
2151
2152
2153
2154
2155
2156
2157
2158
2159
2160
2161
2162
2163
2164
2165
2166
2167
2168
2169
2170
2171
2172
2173
2174
2175
2176
2177
2178
2179
2180
2181
2182
2183
2184
2185
2186
2187
2188
2189
2190
2191
2192
2193
2194
2195
2196
2197
2198
2199
2200
2201
2202
2203
2204
2205
2206
2207
2208
2209
2210
2211
2212
2213
2214
2215
2216
2217
2218
2219
2220
2221
2222
2223
2224
2225
2226
2227
2228
2229
2230
2231
2232
2233
2234
2235
2236
2237
2238
2239
2240
2241
2242
2243
2244
2245
2246
2247
2248
2249
2250
2251
2252
2253
2254
2255
2256
2257
2258
2259
2260
2261
2262
2263
2264
2265
2266
2267
2268
2269
2270
2271
2272
2273
2274
2275
2276
2277
2278
2279
2280
2281
2282
2283
2284
2285
2286
2287
2288
2289
2290
2291
2292
2293
2294
2295
2296
2297
2298
2299
2300
2301
2302
2303
2304
2305
2306
2307
2308
2309
2310
2311
2312
2313
2314
2315
2316
2317
2318
2319
2320
2321
2322
2323
2324
2325
2326
2327
2328
2329
2330
2331
2332
2333
2334
2335
2336
2337
2338
2339
2340
2341
2342
2343
2344
2345
2346
2347
2348
2349
2350
2351
2352
2353
2354
2355
2356
2357
2358
2359
2360
2361
2362
2363
2364
2365
2366
2367
2368
2369
2370
2371
2372
2373
2374
2375
2376
2377
2378
2379
2380
2381
2382
2383
2384
2385
2386
2387
2388
2389
2390
2391
2392
2393
2394
2395
2396
2397
2398
2399
2400
2401
2402
2403
2404
2405
2406
2407
2408
2409
2410
2411
2412
2413
2414
2415
2416
2417
2418
2419
2420
2421
2422
2423
2424
2425
2426
2427
2428
2429
2430
2431
2432
2433
2434
2435
2436
2437
2438
2439
2440
2441
2442
2443
2444
2445
2446
2447
2448
2449
2450
2451
2452
2453
2454
2455
2456
2457
2458
2459
2460
2461
2462
2463
2464
2465
2466
2467
2468
2469
2470
2471
2472
2473
2474
2475
2476
2477
2478
2479
2480
2481
2482
2483
2484
2485
2486
2487
2488
2489
2490
2491
2492
2493
2494
2495
2496
2497
2498
2499
2500
2501
2502
2503
2504
2505
2506
2507
2508
2509
2510
2511
2512
2513
2514
2515
2516
2517
2518
2519
2520
2521
2522
2523
2524
2525
2526
2527
2528
2529
2530
2531
2532
2533
2534
2535
2536
2537
2538
2539
2540
2541
2542
2543
2544
2545
2546
2547
2548
2549
2550
2551
2552
2553
2554
2555
2556
2557
2558
2559
2560
2561
2562
2563
2564
2565
2566
2567
2568
2569
2570
2571
2572
2573
2574
2575
2576
2577
2578
2579
2580
2581
2582
2583
2584
2585
2586
2587
2588
2589
2590
2591
2592
2593
2594
2595
2596
2597
2598
2599
2600
2
```

Welche Methode hat den komplexesten Kontrollflussgraphen?

Antwort: _____

Für welche Methode wurde die Analysierbarkeit in den Qualitätsbereich FAIR eingestuft?
Warum wurde diese Methode schlecht bewertet? Nutzen Sie den Kiviatgraph und das Qualitäts-Modell um es herauszufinden.

Antwort: _____

Ihre Anwendung erhält in der Regel die Bewertung "Excellent" oder "Good". Ist Ihr Code wirklich so gut?
Tipp: Was müssten Sie noch implementieren, um die Anwendung wirklich einsatzfähig zu machen und wie würde sich das auf die Metriken auswirken?

Antwort: _____
