Fragen zu Software Engineering allgemein:

Was ist UML? Nennen Sie die wichtigsten UML-Diagramme!

UML = **U**nified **M**odeling **L**anguage

Die Unified Modeling Language (UML) ist eine Sprache und Notation zur Spezifikation, Konstruktion, Visualisierung und Dokumentation von Modellen für Softwaresysteme.

Es handelt sich um eine Vereinheitlichung der grafischen Darstellung und Semantik der Modellierungselemente – es wird jedoch keine Methodik beschrieben.

Mit ihrer Hilfe lassen sich Analyse und Design im Softwareentwicklungsprozess beschreiben.

Grundsätzlich werden zwei Hauptdiagrammtypen unterschieden: Strukturdiagramme und Verhaltensdiagramme.

Strukturdiagramme

- Klassendiagramm (Klassen, Beziehungen untereinander, Merkmale)
- Komponentendiagramm (Komponenten, Verdrahtung)
- Verteilungsdiagramm (Verteilung von Artefakten auf Knoten)
- Profildiagramm (Stereotypen)
- Objektdiagramm (Exemplare von Klassen mit beispielhaften Werten)
- Kompositionsstrukturdiagramm (Innerer Aufbau von Klassen oder Objekten)
- Paketdiagramm (Ablagestruktur, Abhängigkeiten)
- Anwendungsfalldiagramm (Anwendungsfälle, Beziehungen zu Akteuren)

Verhaltensdiagramme

- Aktivitätsdiagramm (Aktionen, Flüsse, Verzweigungen)
- Interaktionsdiagramm (zulässige Reihenfolge von Aktivitäten)
    - Interaktionsübersicht (Übersicht über mehrere Interaktionen)
    - Kommunikationsdiagramm (Topologie des Nachrichtenaustauschs)
    - Sequenzdiagramm (Reihenfolge des Nachrichtenaustauschs)
    - Zeitdiagramm (zeitlicher Ablauf des Nachrichtenaustauschs)
- Zustandsdiagramm (Zustände, Übergänge, Verhalten)
    - Protokollautomat (Zulässige Reihenfolge von Operationen)

Anwendungsfalldiagramm

Ein Anwendungsfalldiagramm entsteht meist während der Anforderungsphase und beschreibt die Geschäftsprozesse, indem es die Interaktion von Personen – oder von bereits existierenden Programmen – mit dem System darstellt. Die handelnden Personen oder aktiven Systeme werden Aktoren genannt und sin im Diagramm als kleine Männchen angedeutet. Anwendungsfälle beschreiben dann eine Interaktion mit dem System.

Klassendiagramm

Für die statische Ansicht eines Programmentwurfs ist das Klassendiagramm einer der wichtigsten Diagrammtypen. Ein Klassendiagramm stellt zum einen die Elemente der Klasse dar, also die Attribute und Operationen, und zum anderen die Beziehungen der Klassen untereinander. Klassen werden als Rechteck dargestellt, die Beziehungen zwischen den Klassen werden durch Linien angedeutet.

Objektdiagramm

Ein Klassendiagramm und ein Objektdiagramm sind sich auf den ersten Blick sehr ähnlich. Der wesentliche Unterschied besteht aber darin, dass ein Objektdiagramm die Belegung der Attribute, als den Objektzustand, visualisiert. Dazu werden sogenannte Ausprägungsspezifikationen verwendet. Mit eingeschlossen sind die Beziehungen, die das Objekt zur Laufzeit mit anderen Objekten hält.

Beschreibt zum Beispiel ein Klassendiagramm eine Person, so ist es nur ein Rechteck im Diagramm. Hat diese Person jedoch Assoziationen zu anderen Personen-Objekten, so können sehr viele Personen in einem Objektdiagramm verbunden sein, während ein Klassendiagramm diese Ausprägung nicht darstellen kann.

Sequenzdiagramm

Das Sequenzdiagramm stellt das dynamische Verhalten von Objekten dar. So zeigt es an, in welcher Reihenfolge Operationen aufgerufen und wann neue Objekte erzeugt werden. Die einzelnen Objekte bekommen eine vertikale Lebenslinie, und horizontale Linien zwischen den Lebenslinien der Objekte beschreiben die Operationen oder Objekterzeugungen. Das Diagramm liest sich somit von oben nach unten.

Beschreiben Sie die wichtigsten Tätigkeiten, die im Zuge eines Software-Projekts anfallen!

Was verstehen Sie unter Anforderungsanalyse?

Welche UML-Diagramme können in der Analysephase verwendet werden?

Was ist ein Sequenzdiagramm und wofür können Sequenzdiagramm eingesetzt werden?

Das Sequenzdiagramm ist ein Vertreter der Verhaltensdiagramme und gehört zur Untergruppe der Interaktionsdiagramme. Es wird nicht nur der momentane Zustand der Objekte abgebildet, sondern ein ganzer Ablauf. Sie zeigen den Ablauf von Objektinteraktionen.

Das Sequenzdiagramm stellt das dynamische Verhalten von Objekten dar. So zeigt es an, in welcher Reihenfolge Operationen aufgerufen und wann neue Objekte erzeugt werden. Die einzelnen Objekte bekommen eine vertikale Lebenslinie, und horizontale Linien zwischen den Lebenslinien der Objekte beschreiben die Operationen oder Objekterzeugungen. Das Diagramm liest sich somit von oben nach unten.

Im Besonderen steht der zeitliche Verlauf der Kommunikation im Vordergrund. Die Zeit läuft von oben nach unten. Sequenzdiagramme dürfen als Beteiligte Objekte und weitere Akteure beinhalten. Die Beteiligten kommunizieren über Nachrichten. Sie werden durch ein Rechteck mit Beschriftung, an dem unten eine vertikale, gestrichelte Linie angebracht ist, dargestellt. Die Gesamtheit aus Rechteck und gestrichelter Linie wird als Lebenslinie bezeichnet.

Rechtecke auf den gestrichelten Linien signalisieren, wann eine Lebenslinie aktiv ist. Methodenaufrufe und weitere Kommunikationsmöglichkeiten der Akteure untereinander werden durch Nachrichten dargestellt, die aus einer Beschreibung und einem waagerechten Pfeil bestehen

Eine synchrone Nachricht besitzt eine schwarze, gefüllte Pfeilspitze und wird mit einer Antwort (ungefüllte Pfeilspitze, gestrichelte Linie) quittiert. Synchron bedeutet in diesem Zusammenhang, dass die aufrufende Lebenslinie so lange blockiert, bis sie eine Antwort bekommt. Die Angabe des Antwortpfeils ist optional. Sie ist aber obligatorisch, wenn die Methode einen Wert zurück liefert.

Eine asynchrone Nachricht wird durch eine ungefüllte Pfeilspitze an einer durchgehenden Linie dargestellt. Asynchron bedeutet, dass der Sender der Nachricht nicht auf eine Antwort wartet und nicht blockiert.

Am besten verwendet man Sequenzdiagramme zur beispielhaften Veranschaulichung von Vorgängen. Sie bilden nichts Statisches ab, sondern etwas Dynamisches. Sequenzdiagramme werden immer dann gebraucht, wenn man Abläufe innerhalb von Softwaresystemen darstellen will. Mit ihnen ist man in der Lage die Kommunikation zwischen Klassen oder Objekten darzustellen.

Welche Tätigkeiten fasst man unter dem Begriff „Design“ zusammen?

Welche Beziehungen zwischen Klassen bzw. Objekten können in einem Klassendiagramm eingetragen werden?

Spezialisierung/Generalisierung

Eine Generalisierung/Spezialisierung ist eine Beziehung zwischen einem allgemeineren und einem spezielleren Element (bzw. umgekehrt), wobei das speziellere weitere Eigenschaften hinzufügt und sich kompatibel zum allgemeinen verhält. Vererbung ist ein Umsetzungsmechanismus für die Relation zwischen Ober- und Unterklasse, wodurch Attribute und Operationen der Oberklasse auch den Unterklassen zugänglich gemacht werden.

Notation: Die Vererbungsbeziehung wird mit einem großen, nicht gefüllten Pfeil dargestellt, wobei der Pfeil von der Unterklasse zur Oberklasse zeigt.

Abhängigkeit

Eine Abhängigkeit ist eine Beziehung von einem (oder mehreren) Quellelement(en) zu einem (oder mehreren) Zielelement(en).

Die Zielelemente sind für die Spezifikation oder Implementierung der Quellelemente erforderlich.

Notation: Dargestellt wird eine Abhängigkeit durch einen gestrichelten Pfeil, wobei der Pfeil vom abhängigen auf das unabhängige Element zeigt.

Realisierung

Die Realisierungsbeziehung ist eine spezielle Abstraktionsbeziehung. Es ist eine Beziehung zwischen einer Implementierung und ihrem Spezifikationselement.

Das abhängige Element implementiert das unabhängige Element (z.B. eine Schnittstelle oder ein abstraktes Element).

Notation: eine gestrichelte Linie mit einer dreieckigen Pfeilspitze,1 wobei der Pfeil vom abhängigen auf das unabhängige Element zeigt.

Assoziation

Eine Assoziation beschreit als Relation zwischen Klassen die gemeinsame Semantik und Struktur einer Menge von Objektverbindungen.

Gewöhnlich ist eine Assoziation eine Beziehung zwischen zwei Klassen. Grundsätzlich kann aber jede beliebige Anzahl von Klassen an einer Assoziation beteiligt sein.

Notation: Assoziationen werden durch eine Linie zwischen den beteiligten Klassen dargestellt. An den jeweiligen Enden der Linie wird die Multiplizität notiert.

Gerichtete Assoziation

Eine gerichtete Assoziation ist eine Assoziation, bei der von der einen beteiligten Klasse zur anderen direkt navigiert werden kann, nicht aber umgekehrt.

Notation: Eine gerichtete Assoziation wird wie eine gewöhnliche Assoziation notiert, jedoch hat sie auf der Seite der Klasse, zu der navigiert werden kann eine offene Pfeilspitze. Die Richtung, in die nicht navigiert werden kann, wird durch ein kleines Kreuz auf der Seite der Klasse markiert, zu der nicht navigiert werden kann.

Attributierte Assoziation

Eine attributierte Assoziation verfügt sowohl über die Eigenschaften einer Klasse als auch über die einer Assoziation. Es kann gesehen werden als eine Assoziation mit zusätzlichen Klasseneigenschaften oder als Klasse mit zusätzlichen Assoziationseigenschaften.

Wenn zwei Klassen in Beziehung zueinander stehen, kann es sein, dass es Eigenschaften gibt, die weder zu einen noch zur anderen Klasse gehören, sondern zur Beziehung zwischen den beiden. Mit einer Assoziationsklasse kann dies modelliert werden.

Notation: Attributierte Assoziationen werden wie gewöhnliche Assoziationen dargestellt, zusätzlich ist jedoch über eine gestrichelte Linie, die von der Assoziationslinie abgeht, ein normales Klassensymbol angehängt.

Mehrgliedrige Assoziation

Eine mehrgliedrige Assoziation ist eine Assoziation, an der mehr als zwei Klassen beteiligt sind.

Notation: Eine mehrgliedrige Assoziation wird mit einer nicht ausgefüllten Raute gezeichnet, die größer ist als die Aggregationsraute. Die Klassen werden mit Linien mit der Raute verbunden.

Qualifizierte Assoziation

Bei einer qualifizierten Assoziation wird die referenzierte Menge der Objekte durch qualifizierende Attribute in Partitionen unterteilt.

Die durch eine Assoziation spezifizierte Menge von verlinkten Objekten kann durch eine ihrer Eigenschaften in Untermengen (Partitionen) aufgeteilt werden. Diese Eigenschaf kann man als Qualifiziere modellieren.

Notation: Das für die Assoziation benutzte qualifizierende Attribut wird in einem Rechteck an der Seite der Klasse notiert, die über diesen Qualifizierer auf das Zielobjekt zugreift.

Aggregation

Eine Aggregation ist eine Assoziation, erweitert um den semantisch unverbindlichen Kommentar, dass die beteiligten Klassen keine gleichwertige Beziehung führen, sondern eine Ganzes-Teile-Hierarchie darstellen. Eine Aggregation soll beschreiben, wie sich etwas Ganzes aus seinen Teilen logisch zusammensetzt.

Kennzeichnend für alle Aggregationen ist, dass das Ganze Aufgaben stellvertretend für seine Teile wahrnimmt. Im Gegensatz zur Assoziation führen die beteiligten Klassen keine gleichberechtigen Beziehung, sondern eine Klasse (das Aggregat) bekommt eine besondere Rolle und übernimmt stellvertretend die Verantwortung und Führung.

Notation: Eine Aggregation wird wie eine Assoziation als Linie zwischen zwei Klassen dargestellt und zusätzlich mit einer kleinen Raute versehen. Die Raute steht auf der Seite des Aggregats (des Ganzen).

Komposition

Eine Komposition ist eine strenge Form der Aggregation, bei der das Ganze verantwortlich ist für die Existenz und Speicherung der Teile. Sie beschreibt, wie sich etwas Ganzes aus Einzelteilen zusammensetzt und diese kapselt.

Notation: Die Komposition wird wie die Aggregation als Linie zwischen zwei Klassen gezeichnet und mit einer kleinen gefüllten Raute auf der Seite des Ganzens versehen.

Was steht in einem Projektplan? Was ist ein Meilenstein?

Welche Methoden der Qualitätssicherung können in einem Software-Projekt eingesetzt werden?

Was versteht man unter Versionskontrolle?

Was beeinflusst den Aufwand eines Software-Projekts? Wie kann Aufwand geschätzt werden?

Welche Vorgehensmodelle der Software-Entwicklung gibt es?