

23 Ziele und Vorgehensweise

23.1 Einführung

In der Anforderungsermittlung wurden die funktionalen Anforderungen an das zu erstellende Anwendungssystem separat aus den drei Blickwinkeln Funktion, Struktur und Verhalten betrachtet und im Anwendungsfallmodell, im Domänenklassenmodell und in Interaktions- und ggf. Zustandsdiagrammen erfasst. Skizzen der Benutzungsoberfläche mit zusätzlichen Angaben zur geschäftsprozessorientierten Navigation sowie die nichtfunktionalen Anforderungen komplettierten die Anforderungsspezifikation.

Die übergeordnete Richtschnur dabei war die Problemadäqutheit der Modelle, da diese – wenn auch in unterschiedlichem Maß – die Basis für Diskussionen mit den Anwendern bilden. Eine an der Problemwelt orientierte »natürliche« Modellierung, wie sie insbesondere im Anwendungsfallmodell erreicht werden kann, kommt der Sichtweise der Anwender am nächsten und ist Voraussetzung insbesondere für die wichtige Validierung der Anforderungen, die nur in engster Zusammenarbeit mit den Anwendern vorgenommen werden kann. Ein weiteres Argument für eine solche »Multimodellspezifikation« ist die Beherrschung der Komplexität durch die klare Trennung der in den unterschiedlichen Modellen darstellbaren Sichten auf die Anforderungen.

Insbesondere bei großen Projekten hat sich gezeigt, dass eine derartig »anwenderzentrierte«, deskriptive, umgangssprachlich dokumentierte und darüber hinaus auf unterschiedliche Modelle verteilte Anforderungsspezifikation keine hinreichende Voraussetzung für die erfolgreiche Realisierung eines Anwendungssystems ist. Es empfiehlt sich in solchen Fällen, als Zwischenschritt zwischen Anforderungsermittlung und den ersten Realisierungsaktivitäten die sog. *Softwarespezifizierung (analysis)* einzuschieben, in der man die funktionalen Anforderungen im Hinblick auf die formale Sprache der Entwickler aufbereitet und präzisiert. Insbesondere sind die in der Anforderungsspezifikation separat betrachteten Aspekte Funktion und Struktur miteinander in einem hinsichtlich der Realisierung präskriptiven Klassenmodell zu integrieren. Dazu transformiert man das Anwendungsfallmodell in Spezifikationsklassen, führt diese mit dem Domänenklassenmodell zusammen und verankert dann die Verhaltenssicht in Operationen des resultierenden Klassenmodells der Softwarespezifikation.

Hauptziele der Softwarespezifizierung sind somit die Umsetzung der (textuell beschriebenen) Anwendungsfälle in Klassen, die mit dem Domänenklassenmodell in einem initialen *Klassenmodell der Softwarespezifikation* zusammengeführt werden, die anschließende Überarbeitung und Präzisierung dieses Klassenmodells, die Verankerung der Verhaltenssicht in Operationen und die *Verifikation* der Softwarespezifikation

(und gleichzeitig auch der Anforderungsspezifikation), bei der die Konsistenz und Vollständigkeit der einzelnen Modelle in sich überprüft und die verschiedenen Modelle gegeneinander abgeglichen werden. Wichtig ist, dass die resultierenden Modelle immer noch »technologieneutral«, also von der Realisierungsplattform unabhängig sind.

23.2 Softwarespezifizierung in Moos

Die Tätigkeiten der Softwarespezifizierung im Rahmen von MOOS werden im Folgenden grob skizziert. Wichtig ist, hierbei nicht streng sequenziell, sondern teilweise iterativ und parallel vorzugehen.

- Das Domänenklassenmodell wird weitgehend strukturgleich als initiales Klassenmodell der Softwarespezifikation übernommen. Gegebenenfalls sind weitere Klassen aufzunehmen, die weiter gehende Anforderungen wie z.B. Mechanismen zur Authentifizierung und Zugriffskontrollen berücksichtigen.
- Die in der Anforderungsspezifikation festgehaltenen Anwendungsfälle und ggf. Geschäftsprozessfunktionen werden in Klassen der Softwarespezifikation transformiert, denen die funktionalen Anforderungen als Verantwortlichkeiten zugeordnet werden. Diese Klassen (und die zugehörigen Assoziationen) werden als *Geschäftsprozess- und Anwendungsfall-Funktionsklassen* neu eingeführt.
- Skizzen und Angaben zur Benutzungsoberfläche aus der Anforderungsspezifikation werden mit dem Klassenmodell der Softwarespezifikation verbunden.
- Die textuellen Beschreibungen der Geschäftsprozessfunktionen und Anwendungsfälle sowie die zur Illustration des ablauforientierten Verhaltens in der Anforderungsermittlung erstellten Interaktionsdiagramme liefern die Basis, um das Klassenmodell der Softwarespezifikation mit Operationen auszukleiden. Hand in Hand damit wird das ablauforientierte Verhalten komplexer Anwendungsfälle mit Kollaborationen und Interaktionsdiagrammen, die auf dem Klassenmodell der Softwarespezifikation basieren, spezifiziert. Falls nötig, werden existierende Zustandsdiagramme verfeinert und/oder neu erstellt.
- Das so erhaltene Klassenmodell der Softwarespezifikation wird gemäß den Heuristiken in Kapitel 25 und den Regeln in Abschnitt 18.4 aufbereitet und bereinigt. Außerdem wird es in Pakete aufgeteilt.
- Abschließend findet eine *Verifikation* statt, bei der man die einzelnen Modelle in sich auf Konsistenz und Vollständigkeit überprüft sowie gegeneinander abgleicht. Zusätzlich können die in der Anforderungsermittlung erstellten Abnahme- und Systemtestfälle präzisiert werden.

Diese Tätigkeiten münden in eine *Softwarespezifikation*, deren Elemente denjenigen der Anforderungsspezifikation zugeordnet sind und diese ergänzen und präzisieren. Zusammen mit den textuellen Beschreibungen der Anwendungsfälle und den nicht-funktionalen Anforderungen bildet die Softwarespezifikation die Vorgabe für die Realisierung des Anwendungssystems. Die Anwendungsfälle spielen nicht zuletzt auch deswegen weiterhin eine wichtige Rolle, weil sie die Grundlage für die Projektplanung und den Abnahmetest bilden.

Die Softwarespezifikation beinhaltet im Kern folgende Teile:

- Das Klassenmodell, dessen Klassen mit ihren Attributen, Schnittstellen und Assoziationen die funktionalen Anforderungen präzise formulieren.
- Interaktions- und Zustandsdiagramme, die das Verhalten des Anwendungssystems auf das der Spezifikationsklassen zurückführen.
- Dokumente, in denen die Durchführung und die Ergebnisse der Verifikationsaktivitäten festgehalten sind.

Zusammengefasst entlastet die Softwarespezifizierung einerseits die vorangehende Anforderungsermittlung von zu formalen und feingranularen Aspekten und liefert andererseits Dokumente, welche die Anforderungsspezifikation hinsichtlich der Realisierung ergänzen, präzisieren und verifizieren.

Bei kleinen Projekten mit weniger komplexen Anforderungen sowie bei einzelnen Iterationen in Vorgehensmodellen wie dem RUP (vgl. Abschnitt 3.3) kann es sinnvoll sein, auf die Softwarespezifizierung als dedizierte Tätigkeit zu verzichten. Man mache sich aber klar, dass eine systematische Softwareentwicklung auf die hier zur Softwarespezifizierung zusammengefassten Aktivitäten nicht völlig verzichten kann, denn eine deskriptive, für den Anwender verständliche Anforderungsspezifikation kann nicht gleichzeitig den präzisen Charakter einer präskriptiven Realisierungsvorgabe haben. Die Softwarespezifizierung kann daher lediglich auf die Anforderungsermittlung und die ersten Realisierungsaktivitäten aufgeteilt, nicht aber völlig außer Acht gelassen werden.