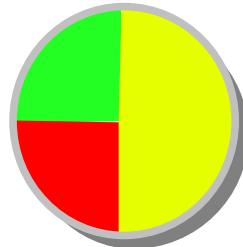


## Fakten zum Thema Softwareentwicklung

### Erfolgsquote

Nur **25 %** aller Softwareprojekte sind erfolgreich!

- **ca. 25 %** aller Softwareprojekte sind erfolgreich, hinsichtlich Zeit-, Budget- und Funktionsvorgaben.
- **ca. 50 %** aller Softwareprojekte liegen über den Zeit- und / oder Budgetvorgaben bzw. erfüllen nicht die geforderte Funktionalität.
- **ca. 25 %** aller Softwareprojekte werden abgebrochen.



Angaben zur Erfolgsquote von Softwareprojekten differieren sehr stark, da sie nicht nur von den Modalitäten der jeweiligen Studie, sondern auch von temporären und regionalen Unterschieden beeinflusst werden.

### Fehleinschätzungen

Das nur 25 % aller Softwarepakete als erfolgreich eingestuft werden können, liegt sehr häufig an dem falsch eingeschätzten Zeitbedarf. Nicht selten wird wesentlich mehr Zeit benötigt als ursprünglich veranschlagt. Das hat natürlich Auswirkungen auf die Projektkosten, so dass ein überzogenes Budget oft eine Folge der zeitlichen Fehleinschätzung ist. Dabei wird der Zeitaufwand



- sowohl bezüglich der **Projektorganisation**,
- als auch für den **Kommunikationsbedarf**
- sowie die eigentliche **Programmierdauer** zu niedrig eingeschätzt.

Bei gut dokumentiertem, zuverlässigem, test- und wartbarem Code produziert ein Programmierer im längerfristigen Durchschnitt

- weniger als **20 LOC** (Lines of Code) pro Arbeitstag.

Auch hierzu gibt es unterschiedliche Studien, die auch zu verschiedenen (zum Teil noch deutlich niedrigeren) Werten kommen. Kurzfristig können natürlich wesentlich mehr als 20 LOC am Tag erstellt werden, aber nach einem halben Jahr Projektlaufzeit wird man effektiv auch auf einen Wert in dieser Größenordnung kommen.

### Teamgrößen

Wenn sich abzeichnet, dass ein Softwareprojekt nicht fristgerecht fertig gestellt werden kann wird häufig versucht den Softwareentwicklungsprozess durch zusätzliche Mitarbeiter zu beschleunigen. Doch bei der Teambildung ist zu beachten, dass die folgende einfache Formel NICHT stimmt!

$$\frac{\text{ProjektdauerInPersonenMonaten}}{\text{MitarbeiterZahl}}$$

Auf Grund des erhöhten Organisations- und Kommunikationsbedarfs wird der tatsächlich zu erwartende Zeitbedarf deutlich darüber liegen! Zudem behindern sich zu viele Mitarbeiter in einem Team eher gegenseitig, so dass es gar keinen Sinn macht die Mitarbeiteranzahl in einem Team

beliebig aufzustocken. Vielmehr ist die optimale Teamgröße abhängig von der Größe des Projektes. Nach Brooks, Rausch-Schott und Retschitzegger gilt folgende Näherungsformel für die optimale Teamgröße:

$$\text{OptimaleMitarbeiterZahl} = \sqrt{\text{ProjektdauerInPersonenMonaten}}$$

Wenn beispielsweise ein Projekt auf 16 PersonenMonate geschätzt wird, wäre die optimale Teamgröße 4 Personen. Dies ist jedoch lediglich ein sinnvoller Vorschlag für die Teamgröße.

## Vorgehensweise

Eine optimale Vorgehensweise bei der Entwicklung von Softwareentwicklungsprojekten gibt es nicht. Einige häufig wiederzufindende Praktiken bzw. Forderungen sind jedoch:

- Klare schriftlich fixierte "Vision" des Projektziels
- Erwarte das Unerwartete: von vornherein Änderungen der Anforderungen einplanen
- Design und Architektur möglichst einfach, gut strukturiert und änderbar
- Intensive kontinuierliche Kundeneinbindung
- Iterativ-inkrementelle Entwicklung mit möglichst kurzen Zyklen, aber inklusive häufigen Integrationstests (möglichst automatisiert)
- Frühe Prototypen
- Frühes und häufiges Testen, eventuell sogar zuerst Testfälle und dann Entwicklung
- Nicht zuviel, aber auch nicht zuwenig Dokumentation

## Qualitätsmerkmale

Sechs Qualitätsmerkmale für Softwareprodukte nach ISO 9126 (DIN 66272)

- **Functionality / Funktionalität:**  
Korrektheit, Angemessenheit, Interoperabilität, Ordnungsmäßigkeit, Sicherheit
- **Reliability / Zuverlässigkeit:**  
Reife, Fehlertoleranz, Wiederherstellbarkeit
- **Usability / Benutzbarkeit:**  
Verständlichkeit, Bedienbarkeit, Erlernbarkeit, Robustheit
- **Efficiency / Effizienz:**  
Wirtschaftlichkeit, Zeitverhalten, Verbrauchsverhalten
- **Maintainability / Wartungsfreundlichkeit:**  
Analysierbarkeit, Änderbarkeit, Stabilität, Testbarkeit
- **Portability / Übertragbarkeit:**  
Anpassbarkeit, Installierbarkeit, Konformität, Austauschbarkeit

