

----- Softwarekrise -----

a) Komplexe, nicht mathematisch beschreibbare Programme für "einfache" Aufgaben

Auf Grund starken Veränderungen im Produktivsystem im Vergleich zu Testumgebungen müssen unendlich viele mögliche Sonderfälle beim Erstellen von Software betrachtet werden. Dadurch werden selbst kleinst-Programme derart aufgeblasen, dass sie deutlich komplexer sind, als sie eigentlich sein müssten.

Beispielsweise, wenn verschiedene Betriebssysteme in config files berücksichtigt werden müssen (Beim entpacken von .zip, .tar etc).

b)

Die momentane reine Entwicklungszeit von Programmen ist nur ein Bruchteil des benötigten Arbeitsaufwands.

Hinzu kommen komplexe, zeitintensive Tests sowie andauernde Betreuung dieser Programme. Vor allem die Betreuung auf unbestimmte Zeit kostet Zeit sowie Humankapital.

Als alternative kann jegliche Software für verschiedene Systeme immer wieder neu programmiert werden.

Dies ist jedoch weder ökonomisch, noch Sinnvoll im Bezug auf Wirtschaftlichkeit der Unternehmen.

Unter Berücksichtigung dieser Punkte sollte jedoch nicht die Qualität der Produkte leiden.

Diese sollen weiterhin in Design, Funktionalität sowie Wartbarkeit die Anforderungen erfüllen.

c)

WINDOWS XP SERVER AN DER RWTH!

Auf Grund veralteter Software und mangelnder Kompatibilität zu zugehörigen Produkten aus diesem Jahrtausend, zieht diese Software die Qualität und Upgrade-Bereitschaft anderer Programme runter.

So wird mittelalterliche Software, meist auf virtualisierten Rechnern, weiterhin zur Verfügung gestellt und dient als einfacher Einstieg, wenn man solch einem Unternehmen z.B. durch Schadcode schaden möchte.

----- Charakterisierung Softwaretechnik -----

a)

„Zielorientierte Bereitstellung und systematische Verwendung von Prinzipien, Methoden und Werkzeugen für die arbeitsteilige, ingenieurmäßige Entwicklung und Anwendung von umfangreichen Softwaresystemen.“

– Lit.: Balzert, S. 36

----- Zu den Kernprozessen von Softwaretechnik gehören: -----

1. Planung

Anforderungsanalyse, Lastenheft, Pflichtenheft, Aufwandsschätzung, Vorgehensmodell

2. Analyse

Auswertung, Mock-Up, Prozessanalyse, Systemanalyse, SA, OOA

3. Entwurf

Softwarearchitektur, SD, OOD, FMC

4. Programmierung

Normierte Prog., Strukturierte Prog., OOP, Funktionale Prog.

5. Validierung und Verifikation

Modultests, Integrationstests, Systemtests, Akzeptanztests

----- Zu den Unterstützungsprozesse von Softwaretechnik gehören: -----

6. Anforderungsmanagement

7. Projektmanagement

Risikomanagement, Projektplanung, Projektverfolgung, Lieferantenvereinbarungen

8. Qualitätsmanagement

SPICE, Incident Management, Problemmanagement, Softwaremetrik, statische Analyse

9. Konfigurationsmanagement

Versionsverwaltung, Änderungsmanagement, Releasemanagement, Releasemanagement,

ITIL

10. Softwareeinführung

11. Dokumentation

Technische Dokumentation, Softwaredokumentation, Bedienungsdokumentation

b)

Durch exakte Planung, daraus resultierender Konzepte und genauer Betrachtung der Anforderungen sollen Eigenschaften der Qualität verbessert werden.

Des Weiteren sollen durch anschließende Tests die Anzahl an möglichen Fehlern (falls möglich auf 0 [Spoiler-Alarm, ist nie möglich]) gesenkt werden.

So soll an den Käufer ein vollständiges, den Anforderungen entsprechendes, möglichst fehlerfreies, effizientes, wartbares und zuverlässiges Produkt geliefert werden.

c)

Stumpf gesagt wirkt sich die korrekte Anwendung sinnvoller Softwaretechnik auf die Planung von Software aus.

So wird sich vor dem Programmieren bei SWT Gedanken über mögliche Fehlerquellen gemacht, und bei dem handwerklich ordentlichen Programmieren erst im Fehlerfall.

Somit ist das stumpfe Programmieren ein kleiner Teilbereich von Softwaretechnik, verpackt mit Planung, Konzeptionierung, sowie anschließender Evaluierung.

----- Git -----

a)

Bei eventuell auftretenden Fehlern oder ungewollten „Features“ kann auf eine ältere Version des Codes zurückgespielt werden.

Des Weiteren kann man sich ältere Versionsstände anschauen, sowie Updates genauer inspizieren.

So sieht man bei einem sogenannten „commit“ meist einen beschreibenden Updategrund und kann im Fehlerfall den schuldigen identifizieren.

b)

Git funktioniert ebenfalls lokal auf dem eigenen System.

Des Weiteren ist git schneller und das Zusammenfügen verschiedener Branches ist Benutzerfreundlicher.

c)

a) strg+z

b) git checkout

c) git reset