

1.– 4. September 2014
in Nürnberg



Herbstcampus

Wissenstransfer
par excellence

Vorsicht Schuldenfalle

Was die IT aus der Finanzwelt lernen kann

Tammo van Lessen (innoQ)
Jörg Nietzsche (T-Systems)

Wer sind wir?

Tammo van Lessen
Senior Consultant bei innoQ
SOA / BPM / Architektur / Web
Apache Member und VP Apache ODE



Dr. Jörg Nitzsche
Senior Architect bei T-Systems
EAM / SOA / BPM / Architektur

Inhalt

- Metaphern
- Arten der Schulden
- Was ist mit den Zinsen?
- Wie verschulde ich mich?
- Was muss ich über Schulden wissen?
- Was tue ich bei drohender Insolvenz?

Metaphern

- Warum?
 - Helfen Dinge deutlich zu machen
- Häufig verwendete Metaphern:
 - Haus = Applikation
 - Bebauungsplanung = Enterprise Architektur Management
- Heute:
 - Natürliche Person = Applikation

Natürliche Person

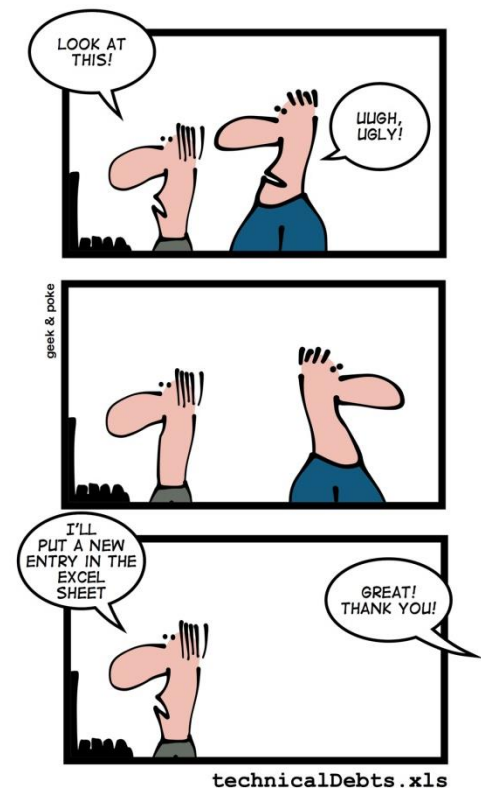
- Spätestens seit Peter Zwegat wissen wir:
 - Personen verschulden sich, können die Schulden nicht tilgen und die Zinsen nicht bezahlen
- Weitere Definition:
 - Schuldenlast = Kosten zur Problemlösung
 - Zinsen = Kosten mit der aktuellen Situation zu leben



(c) RTL/Gregorowius

Technische Schulden

- Metapher geprägt von Ward Cunningham '92
- Diskutiert von Martin Fowler & Steve McConnell
- In der Praxis viel zu wenig ernst genommen



1.– 4. September 2014
in Nürnberg



Herbstcampus

Wissenstransfer
par excellence

Schulden

Was für Arten von Schulden gibt es?

Design & Architektur

- Ungeeigneter Architekturstil
- Falsche Wahl der Werkzeuge
- Ignorieren von Konzernstandards
- Zu viele Kompromisse und Workarounds
- Frameworks „vergewaltigt“



Code-Qualität

- Hohe Anzahl von Ausfällen / Fehlern
- Hohe Anzahl kritischer Bugs
- Alarmierende Code-Metriken
- Code-Duplizierung
- Kein „clean code“ / SOLID

Testen

- Fehlende Automatisierung
- Manuelle Regressionstests



Dokumentation

- Wissen wird nicht dokumentiert.
- Wissen wird nicht geteilt.
- Wissen wird nicht verteilt.

- „Bus factor“

Umgebung

- Wackelige Infrastruktur
- Unflexibler Betrieb
- Unangemessene Konzernstandards
- Fehlerhafte Komponenten
- Unzuverlässige Umsysteme

1.– 4. September 2014
in Nürnberg



Herbstcampus

Wissenstransfer
par excellence

Zinsen

Was ist mit den Zinsen?

Wie hoch sind die Zinsen?

- Betrieb:
 - Neustart des Servers jeden Tag wegen Memory Leak
- Entwicklung:
 - Aufgrund einer Vorgabe muss EJB 2.1 statt 3.0 verwendet werden und deswegen entsteht viel mehr Aufwand
 - Aufgrund fehlender automatisierter Regressionstests sind aufwendige manuelle Tests notwendig

Was muss ich bzgl. der Zinsen wissen?

- Generell muss man beim Schulden machen wissen, wie hoch die Folgekosten sind:
 - Bsp.: USA Finanzkrise / Immobilienkrise
 - Es wurden Darlehen mit variablen Zinsen vergeben, d.h. die Schuldner kannten die Kosten nicht

Gehen Sie nur kalkulierbare Schulden ein!

1.– 4. September 2014
in Nürnberg



Herbstcampus

Wissenstransfer
par excellence

Schuldenaufnahme

Wie nehme ich Schulden auf?

„Pech“

- Unvorhersehbar
- Unbeeinflussbar
- Applikation:
 - „Die strategische Ausrichtung hat sich geändert und das Framework XY darf nicht mehr verwendet werden“
 - „Das Produkt wurde von einem anderen Anbieter übernommen und wird nicht mehr supported / zu gänzlich anderen Konditionen (höhere Zinsen).
- Natürliche Person:
 - „Die Gesetzgebung hat sich geändert, und es darf nicht mehr mit Gas geheizt werden. Die Heizung muss erneuert werden“

Unachtsamkeit

- Verantwortungsloses / „Krankhaftes“ Verhalten
- Mangelnde Kompetenz

- Applikation:
 - „Featuresucht“
 - „Es gibt Logging-Frameworks? Das hätte uns 3 Monate Entwicklungszeit gespart.“
 - „Oh, das Framework das wir ausgesucht habe skaliert ja gar nicht entsprechend unseres Mengengerüsts.“
- Natürliche Person:
 - „Spielsucht“
 - „Kaufrausch“
 - „Ach, die Reisekosten muss ich selber bezahlen...“

Strategisch

- Geplant
- Langfristig
- Applikation:
 - „Wir wollen mit unserem Produkt die ersten auf dem Markt sein, koste es was es wolle.“
- Natürliche Person:
 - „Baudarlehen“

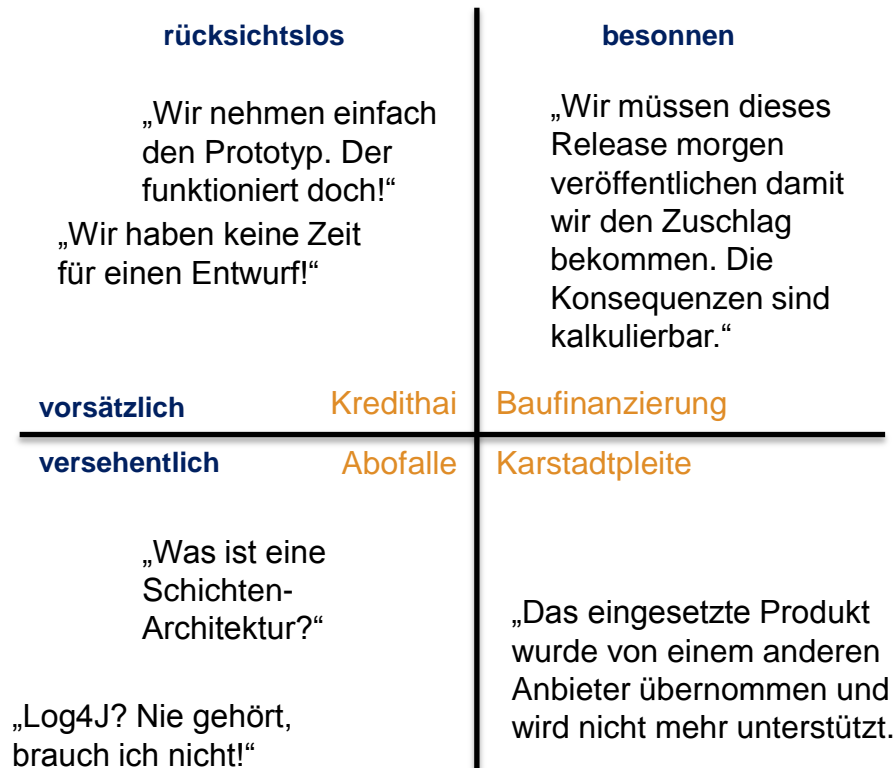
Taktisch

- Geplant
- Kurzfristig
- Applikation:
 - „Wir brauchen das Feature im nächsten Release, können es aber bis dahin nicht sauber umsetzen. Wir fixen das später.“
 - „Die Applikation unterstützt eine Aktion und muss nur 6 Monate laufen, wie ist egal“
- Natürliche Person:
 - „Kredit zur Weihnachtszeit, bis zur nächsten nächsten Bonuszahlung“
 - „Knockin‘ on heavens door“

Inkrementell

- Sammlung vieler kleiner Kompromisse
- Mangelnde Übersicht
- „Ich weiß grad nicht wie viele Ratenkredite ich schon habe, aber den einen werd' ich schon noch verkraften.“
- „Kreditkartenschulden“

Kategorisierung in Quadranten



Prognose bei Verschuldung

- Rücksichtslose Verschuldung:
 - Zinszahlungen werden so hoch, dass sie nicht mehr zurückbezahlt werden können, bis hin zur Insolvenz
- Besonnene Verschuldung:
 - In der Regel gut geplant und überlegt, d.h. in der Regel kann der Kredit zurückbezahlt werden, nur nicht bei unvorhersehbaren Ereignissen.

Treffen Sie bewusste Entscheidungen, versuchen Sie unvorhergesehene Ereignisse zu antizipieren

1.– 4. September 2014
in Nürnberg



Herbstcampus

Wissenstransfer
par excellence

Was muss ich noch über Schulden
wissen?

Ursachen für Schulden

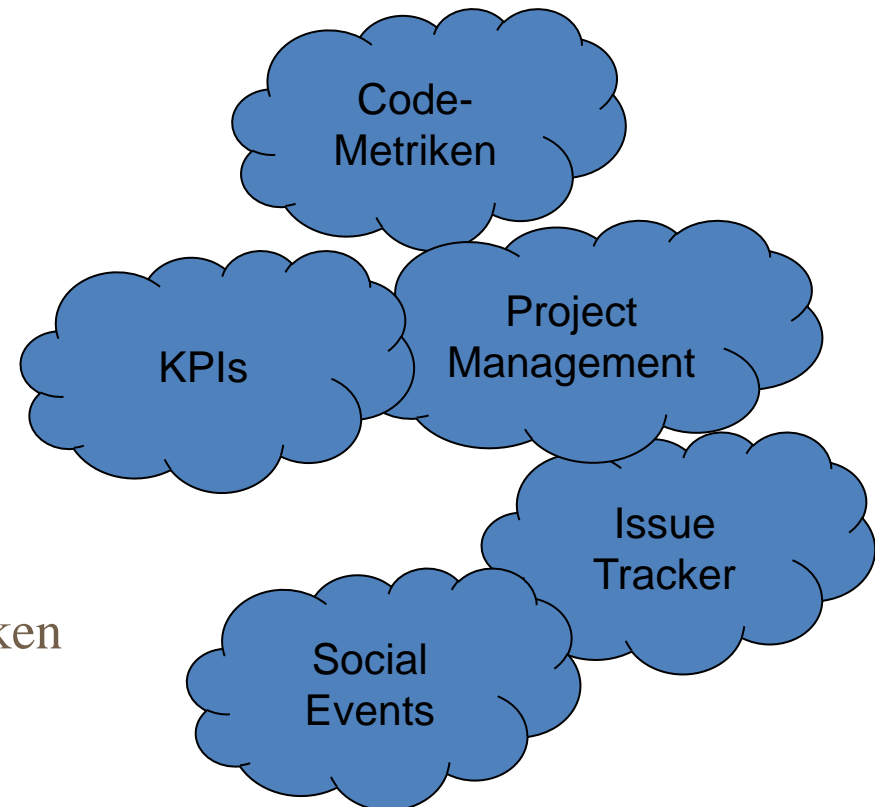
- Mangelnde Erfahrung aller Beteiligten:
 - Projektleiter / Applikationseigner
 - Architekt
 - Entwickler
 - ...
- Überlastung der Beteiligten:
 - Entwickler in Helpdesk-Aufgaben eingebunden.
 - Zu ambitionierte Planung
- Unklare und sich ändernde Anforderungen
- Oft auch Kulturprobleme:
 - Gegeneinander statt miteinander
 - Verschiedene Stakeholder mit unterschiedlichen Interessen

Wie erkenne ich meine Schulden?

- Im realen Leben
 - Kontoauszug
 - Kreditkarte wird eingezogen
 - Schreiben vom Gerichtsvollzieher
 - Besuch des Gerichtsvollziehers
 - Haus wird zwangsversteigert

Wie erkenne ich meine Schulden?

- In der IT
 - Sinkende Qualität
 - Sinkende Produktivität
 - Sinkende Testüberdeckung
 - Steigende Anzahl Defects
 - Steigende Testaufwände
 - Steigende Anzahl Hacks
 - Verspätete Releases
 - Stress vor Deadlines
 - Angst vor Code-Änderungen
 - Verwendung veralteter Bibliotheken



Was mache ich mit meinen Schulden?

- Schulden so belassen und einfach nur die Zinsen zahlen
- Schulden abzahlen
 - Ansätze:
 - 10% der Teamzeit für Refactorings und Aufräumarbeiten reservieren.
 - Aufräum-Releases planen.
 - Einen Schulden-Backlog pflegen
 - Mit Schuldenlast und Zinssatz

Wer hält mich davon ab Schulden zu machen?

- Natürliche Person:
 - Erziehungsberechtigter / Ich selber
 - Gläubiger: Banken
 - Schufa
- Und meine Applikation?
 - Applikationseigner / Projektleiter?
 - Budgetverantwortliche / Fachbereich?
 - Enterprise Architekt bei QG Architecture
 - Architekt

Wie vermeiden Architekt und Entwickler Schulden?

- Kontinuierliches Refactoring
- Automatisierung (Testen, Deployment,...)
- Geteiltes und verteiltes Wissen
- Clean Code / SOLID
- Früh scheitern
- Auf hohes, einheitliches Niveau achten
- Pfadfinderregel

Wie können Sie unterstützt werden?

- Management (Projektleiter / Applikationseigner)
 - Technische Schulden erfassen und ernst nehmen (Schuldenmonitor)
 - Indizien und Kennzahlen nicht den Entwicklern zur Last legen (Unternehmenskultur verbessern)
- Budgetverantwortliche / Fachbereich
 - Technische Schulden ernst nehmen
 - Raum für Schuldenabbau geben
 - Requirements klar definieren

1.– 4. September 2014
in Nürnberg



Herbstcampus

Wissenstransfer
par excellence

Was passiert, wenn mich die
Schulden erdrücken?

Was passiert bei Insolvenz?

- Natürliche Person:
 - Privatinsolvenz anmelden:
 - Verhandlung mit Gläubigern über Schuldenerlass
 - Wenn Verhandlung nicht erfolgreich: Mehrere Jahre am Existenzminimum
- Applikation:
 - Vor der Insolvenz: Versuchen sich mit dem Gläubiger zu einigen
 - Schulden eliminieren / Vorgaben ändern lassen:
 - z.B. DB2 ist strategisch einzusetzen, Oracle wird aber verwendet
 - z.B. Hibernate darf nicht eingesetzt werden, wird aber eingesetzt
 - Höheres Budget für Zinszahlungen (Run-Budget oder Development-Budget)
 - Wenn Verhandlung nicht erfolgreich: Es werden keine neuen fachlichen Anforderungen umgesetzt, bis die neue Applikation schuldenfrei aufgebaut wurde

Wann melde ich Insolvenz an?

- Natürliche Person:
 - Alle Schulden müssen auf den Tisch
 - Danach dürfen keine neuen Schulden bekannt werden
 - Nur wenn man nach Ablauf der Frist auf eigenen Beinen stehen kann

Was bedeutet das für meine Applikation?

- Wenn die Applikation am Ende ist, dann bringt es nichts, Insolvenz der Applikation anzumelden, und eine neue Applikation zu bauen.
- Man muss wie im echten Leben danach wieder auf die Beine kommen können.
- Man muss also sicher sein, dass es beim nächsten Mal besser wird. Also:
 - Keine unkontrollierten unerfahrenen Entwickler, die versehentlich Schulden machen
 - Architektur basierend auf den richtigen Prinzipien
 - Die fachlichen funktionalen und nicht-funktionalen Anforderungen müssen bekannt und sinnvoll sein

Nehmen Sie im Fall der „technischen“ Insolvenz keine fachlichen Schulden mit !

Was tue ich (Entwickler/Architekt) im Projektalltag zur Vermeidung der Insolvenz?

- Ich identifiziere technische Schulden.
- Ich muss meinem Projektmanager und den Stakeholdern klar machen, dass das Projekt in große Schwierigkeiten kommt, wenn die Schulden nicht abgebaut werden.
- Ich bekomme von ihm einen Tag pro Woche oder vollständige Releases für Refactorings (Schuldenabbau)
- Ich muss überwachen, ob das auch wirklich die Schulden verringert
- Wenn ich merke, dass ich die Schulden nicht aus eigener Kraft abbauen kann, dann: haben wir zwei Möglichkeiten:
 - 1) Insolvenz: Applikation durch Neuentwicklung ablösen
 - 2) Irgendwoher billig Geld leihen: höheres Budgets für Run und Development bekommen

1.– 4. September 2014
in Nürnberg



Herbstcampus

Wissenstransfer
par excellence

Zusammenfassung

Zusammenfassung

- Die Metapher „Technische Schulden“ macht das Problem für Entscheider anschaulich und sensibilisiert das Team
- Das hilft ein Umfeld zu schaffen in dem Schulden ernst genommen werden
- Dadurch werden weniger Schulden aufgenommen und ggf. Schulden abgebaut.
- Technische Schulden können guten Gewissens aufgenommen werden.
 - So lange man sie zurückzahlen kann!



- Offene Sammlung von Mustern zur
 - Analyse
 - Evaluierung
 - Verbesserung
von existierender Software
- Initiiert von Gernot Starke
- Lesen, profitieren, mitmachen.
- <http://www.aim42.org/>



Vielen Dank für Ihre Aufmerksamkeit



Dr. Jörg Nitzsche
Senior IT Architect

T-Systems International GmbH
Address: Fasanenweg 5
70771 Leinfelden-Echterdingen, Germany

Contacts: Phone +49 711 999 - 6516
Mobile +49 170 3377528
E-mail joerg.nitzsche@t-systems.com

Bildreferenzen

- Hammer & Schraube:
<https://www.flickr.com/photos/justinbaeder/5317820857/>

Über innoQ

- Beratungsunternehmen für Software-Architekturen
- Strategische IT-Beratung, Architekturconsulting, Entwicklung
- Rationelle Software-Produktion (MDA, MDSD, Metadaten-Integration)
- Service-orientierte Architekturen (SOA) (REST, OSS-Lösungen, Governance)
- Prozess-orientierte Architekturen (BPM)