

# Die NUMMI-Fabrik – eine Parabel über Softwareentwicklung

Continuous Architecture 15.04.2019 08:41 Uhr – Eberhard Wolff – 173 Kommentare

Softwareentwicklung ist nicht wegen der technischen Herausforderungen schwierig, sondern weil so viele Menschen daran beteiligt sind. Ein Blick über den Tellerrand zeigt, wie andere Branchen mit diesen Herausforderungen umgehen und was sie dabei gelernt haben.

Als Beispiel dient [eine Geschichte über eine Autofabrik](#). General Motors (GM) eröffnete die Fabrik 1962 als [Fremont Assembly](#). Leider erwarb sie sich schnell einen katastrophalen Ruf: Die Arbeiter sollen die schlechtesten in der gesamten US-Automobilindustrie gewesen sein. Alkohol war genauso an der Tagesordnung wie unentschuldigtes Fehlen, sodass manchmal die Produktion gar nicht anlaufen konnte. Ebenso sabotierten die Arbeiter die Produktion. Dem Management war die Perspektive der Arbeiter egal. Und natürlich ging es wie so oft in der Massenproduktion um Menge und nicht um Qualität.

## Toyota-Produktionssystem

1984 wurde die gesamte Belegschaft gefeuert. GM und Toyota gründeten in der Fabrik ein Joint Venture. Natürlich musste man Arbeiter einstellen, was schwierig war. Am Ende arbeiteten dann 85 Prozent der alten GM-Belegschaft in der Fabrik. Einige von ihnen schickte Toyota nach Japan, damit sie dort das [Toyota-Produktionssystem](#) erlernten. Am Ende erreichte das Werk dieselbe Produktivität und Fehlerzahl wie die Werke von Toyota in Japan und damit deutlich bessere Ergebnisse als GM.

Das Toyota-Produktionssystem orientiert sich an Werten wie kontinuierlicher Verbesserung, Respekt gegenüber Menschen und langfristigem Denken. Im NUMMI-Werk sind die Werte auch direkt umgesetzt worden: Der Fokus lag auf der Arbeit im Team und auf Qualität. Alle Mitarbeiter – egal ob sie im Management oder in der Produktion arbeiteten – trugen dieselbe Uniform, hatten dieselben Parkplätze und eine gemeinsame Cafeteria. Ebenso gab es keine Entlassungen, ein Vorschlagsprogramm und Entscheidungen im Konsens.

## Jidōka

Ein wichtiger Teil des Toyota-Produktionssystems ist [Jidōka](#). Es steht für "Autonomation", eine autonome Automation oder auch "intelligente Automation" beziehungsweise "Automation mit menschlichem Touch". Bei Anomalien stoppt die Maschine. Das Ziel ist keine vollständige Automatisierung. Dann müssten die Maschinen selber mit Anomalien umgehen können. Das ist aber kaum kosteneffizient realisierbar.

Stattdessen identifiziert der Arbeiter das Kernproblem und beseitigt es. Dazu kann er die Produktionslinie stoppen und die Hilfe der Kollegen anfordern. Produktionsstillstand kann erhebliche Kosten verursachen. Natürlich gibt es Maßnahmen, um so etwas zu vermeiden, aber die Möglichkeit zum Stoppen der Produktion zeigt eindeutig, wie wichtig für das Ziel der Qualität ist.

## Wie es weiterging ...

1998 hatte GM es trotz des Joint Ventures nur in Ausnahmefällen geschafft, die Ideen des Toyota-Produktionssystems in einem anderen Werk umzusetzen. 2009 beendete GM dann das Joint Venture, und 2010 schloss Toyota schließlich die Fabrik.

Das hört sich zwar wie ein Ende an, aber kurz darauf eröffnete Tesla dort sein Werk und bis heute ist das Werk in der Hand von Tesla. Und so geht von diesem Werk heute die Elektro-Mobilitätsrevolution aus.

### ... und die Moral der Geschichte

Aus dieser Geschichte lassen sich mehrere Lehren für die Softwareentwicklung ableiten:

- Basis des Toyota-Produktionssystems ist eine andere Kultur. Sie führt zu einer besseren Qualität, Produktivität und Reduktion von Lagerbeständen. Einige Autohersteller haben versucht, die Qualität direkt zu managen, und haben Ursache und Wirkung verwechselt. Die Kultur führt zum Erfolg des Toyota-Produktionssystems. Das ergibt die bessere Qualität. Für Softwareentwicklungsprozesse wie Agilität oder Continuous Delivery gilt Ähnliches: Wenn sich die Kultur ändert, ergeben sich Vorteile wie Produktivität oder Fehlerfreiheit als Folge. Ein Fokus auf Techniken verstellt diesen Blick oft.
- Es ist schwierig, eine Kultur auf andere Organisationseinheiten zu übertragen. GM hat es trotz des Joint Venture praktisch nicht geschafft, den Erfolg in den eigenen Fabriken zu kopieren. Aber auf der anderen Seite können Personen, die vorher in einer anderen Kultur gearbeitet haben, eine neue Kultur erlernen und umsetzen. So konnte die NUMMI-Belegschaft sich von einer sehr niedriger Produktivität zu einer ausgesprochen hohen Produktivität weiterentwickeln. Prinzipiell ist es also denkbar, die Kultur eines Software-Teams grundlegend zu ändern und so die Ergebnisse zu verbessern.
- Jidōka zeigt ein wichtiges Element des Toyota-Produktionssystems. Fehler sollen offensichtlich werden und dann vollständig beseitigt werden. Maßnahmen wie Agilität oder Continuous Delivery zielen darauf ab, möglichst oft und schnell Software und Features zu testen und in Produktion zu bringen. Dadurch werden Fehler schneller offensichtlich, die dann beseitigt werden können. Wenn also Probleme durch den Prozess offensichtlich werden, ist das ein erwartetes Ergebnis und es gibt die Möglichkeit, darauf zu reagieren und so besser zu werden.
- Gleichzeitig zeigt Jidōka auf, dass eine perfekte Automatisierung als Ziel nicht unbedingt sinnvoll ist. Das gilt auch für Continuous Delivery. Wichtig ist aber, wie man mit Problemen bei der Automatisierung umgeht und die Probleme dann beseitigt.
- Ebenso verdeutlicht Jidōka, welche positiven Effekte es hat, wenn jeder Mitarbeiter durch passende Maßnahmen an Zielen wie Qualität mitarbeiten kann und dabei zur Not auch die Produktion anhalten kann. Es geht dabei nicht nur darum, dass der Mitarbeiter so effektiver Probleme lösen kann. Sondern das Unternehmen drückt so auch Vertrauen in den Mitarbeiter aus, was sich positiv auf die Motivation niederschlägt. Solche Maßnahmen sind auch in der Softwareentwicklung denkbar.

Die NUMMI-Fabrik ist kein zufällig gewähltes Beispiel. Agile Softwareentwicklung orientiert sich an Lean Production und dem Toyota-Produktionssystem. Die wesentlichen Konzepte hat die Agilität aus diesem Bereich übernommen, sodass sich ein Blick auf Beispiele aus der Produktion lohnen.

### tl;dr

Die NUMMI-Fabrik zeigt, dass Kultur entscheidend für den Erfolg eines Unternehmens sein kann, aber diese auch schwer zu ändern ist. Das dort praktizierte Toyota-Produktionssystem ist genauso wie agile Softwareentwicklung im Wesentlichen eine Änderung der Kultur.

### Kommentare lesen (173 Beiträge)

Themenseiten:

AGILE SOFTWAREENTWICKLUNG

CONTINUOUS DELIVERY

SOFTWAREENTWICKLUNG

Anzeige

Phishing-Schutz dank künstlicher Intelligenz  
Interesse an Content Marketing?  
Tipps: Telefonanlagen-Kauf für Unternehmen  
So gehen Sie in der Informationsflut nicht unter  
Acer-Monitore: Ausführliche Tests von Heise-Usern  
Heise RegioConcept – lokales SEO  
Schulung und Simulation: Fit gegen Phishing-Mails  
Endpoint Security: Was ist nötig, was zwecklos?  
SANS: Industrielle Kontrollsysteme absichern!  
E-Mail-Security: Einfallstor Nr.1 absichern!



Eberhard Wolff

Eberhard Wolff (@ewolff) arbeitet als Fellow bei der INNOQ. Er ist seit mehr als 15 Jahren als Architekt und Berater tätig - oft an der Schnittstelle zwischen Business und Technologie. Sein technologischer Schwerpunkt liegt auf modernen Architektur-Ansätzen - Cloud, Continuous Delivery, DevOps, Microservices oder NoSQL spielen oft eine Rolle.

Vita lesen »

News	Rubriken	Blogs	Podcasts	Videos	Konferenzen	Termine
7-Tage-News	Sprachen	Der Dotnet-Doktor	Mein Scrum ist kaputt			
News-Archiv	Architektur/Methoden	the next big thing	SoftwareArchitekTOUR			
	Werkzeuge	Neuigkeiten von der Insel				
	Know-how	Tales from the Web side				
	Standards	Continuous Architecture				
	Literatur	Der Pragmatische Architekt				
	Videos	ÜberKreuz				
	Veranstaltungsberichte	Modernes C++				
		colspan				
		"Ich roll" dann mal aus"				