



**Eberhard Wolff** ✓ • You

Head of Architecture and Co-owner, SWAGLab

6d • 🌐



Mich interessiert:

Wie erreicht man eine flexible Software-Architektur?

- Technische Maßnahmen wie Microservices oder hexagonale Architekturen
- Geschickte fachliche Architektur
- Organisatorische Maßnahmen

Oder vielleicht fällt Euch noch was anderes ein - dann bitte kommentieren.

Bin auf Eure Meinungen gespannt!

### Wie erreicht man eine flexible Software-Architektur?

You can see how people vote. [Learn more](#)

Technik

6%

Gute fachliche Architektur

61%

Organisatorische Maßnahmen

20%

Anderes (bitte kommentieren)

13%

# **20% Rabat auf meine iSAQB-FLEX-Trainings**

**Online 14.-16.Oktober**

**FLEX\_EBERHARD\_10\_25**

**Stuttgart 1.-3. Dezember**

**FLEX\_EBERHARD\_12\_25**

**Online 16.-18. Dezember**

**FLEX\_EBERHARD\_12\_25**

<https://www.socreatory.com/de/trainings/flex>





**Rainer Feike** • 1st

3d ...

Technical Tribe Lead Banking / Senior IT Architect & Developer

Das ganze Thema „flexibel“ braucht ein Redesign.

Ein Arzt gibt auch keine Arm-Prothese, die nächstes Jahr als Bein verwendet werden kann. Und ein Ingenieur baut kein Auto das fliegt, wenn man Flügel dran schraubt.

Unsere Disziplin ist sehr jung. Wir müssen die Theorien schärfen, dazu gehört es, Schluss zu machen, mit dem Anspruch Software beliebig und jederzeit verändern zu können.

Mörtel ist auch flexibel, aber in Grenzen.



**Sven Müller** • 1st

Geschäftsführer bei synyx | Architekt | DDD-Enthusiast | Wir helfen Unternehm...

6d ...

Das geht in drei einfachen Schritten:

- 1.) Den Elfenbeinturm-Architekten rauswerfen.
- 2.) DDD.
- 3.) Akzeptieren, dass Architektur lebendig ist und niemals „abgeschlossen“.

Like · 🗨️ 3 | Reply · 5 replies



**UrsENZler** ✓ • 1st

Software Architekt und Unternehmensentwickler bei Calitime AG Confe...

6d ...

**Sven Müller** nicht dass DDD schlecht wäre, aber wieso fördert es Flexibilität?

Like | Reply



**Sven Müller** • 1st

Geschäftsführer bei synyx | Architekt | DDD-Enthusiast | Wir helfen Unt...

6d ...

**UrsENZler** Zwei Gründe:

Erstens erlauben saubere Domänen-Schnitte, dass einzelne Module bzw. Services von unabhängigen Teams entwickelt und getrennt voneinander skaliert oder auch komplett ausgetauscht werden können.

Zweitens - und das ist meiner Meinung nach der stärkere Grund - sorgt eine kollaborative Herangehensweise dafür, dass die Architektur von allen relevanten Bereichen der Organisation mitgetragen wird. Dadurch gibt es weniger blinde Flecken und ein besseres Verständnis auf allen Ebenen, wie neu entstehende Anforderungen umgesetzt werden können.

Like · 🗨️ 1 | Reply



**Daniel Novak** ✓ • 2nd

Software Engineer | Spar

6d ...

**Sven Müller** DDD ist auch nicht in jedem Business-Kontext die ultimative Lösung. Ist man hauptsächlich im Enterprise Integration unterwegs gibt es andere Design Prinzipien die angebracht sind.

Like · 🗨️ 1 | Reply



**Sven Müller** • 1st

Geschäftsführer bei synyx | Architekt | DDD-Enthusiast | Wir helfen Unt...

5d ...

**Daniel Novak** klar, auch DDD ist keine Silver Bullet, die gibt es einfach nicht im echten Leben 🙌. Weil es bei DDD aber eben darum geht, dass Teams möglichst unabhängig voneinander Mehrwert schaffen können, glaube ich schon, dass man damit nicht ganz falsch liegt, wenn Flexibilität gefordert ist.

Like | Reply



**Rainer Feike** • 1st

Technical Tribe Lead Banking / Senior IT Architect & Developer

3d ...

**Sven Müller** stimmt, ist dann aber keine Lösung sondern Resignation

Like | Reply

Collapse replies



**Jan Rothkegel** ✓ • 2nd

Solution Architect at Infineon Technologies

5d ...

Hier wurden schon viele wertvolle Anhaltspunkte geliefert. In meinem Umfeld entwickeln wir Inhouse-Software für die eigene Fertigungslandschaft. Flexibilität kann hier auch heißen: Ich möchte in der Lage sein inzusourcen, outzusourcen und intern sowie extern umzusourcen. Dabei spielen dann der Technologiestack, Kommunikation (auch heute noch die Sprache) und ein gewisser Grad an Dokumentation eine Rolle. Modularisierung und die Fähigkeit, Entwicklungs- und komplexe Ausführungsumgebungen einfach bei Entwicklern hochzuziehen (Stichwort Docker), sind ein weiterer Pluspunkt.



**Ingo Eichhorst** ✓ • 1st

AI Architect | CTO | Tech Trainer at IONOS & Universities

5d ...

Flexibilität ist eine Illusion. Es ist ja der Sinn von Software Ordnung und Struktur zu liefern. Wir können zwar vorausdenken, was sich in Zukunft evtl. ändern wird und dafür Vorkehrungen treffen, aber dann sind wir ja nicht mehr flexibel, sondern haben gut geplant.

Like · 🗨️ 1 | Reply · 3 replies



**Ralf D. Müller** • 1st

Claude.ai Operator :-) @ DB Systel GmbH | iSAQB® CPSA®-A Certified...

5d ...

**Ingo**, Flexibilität ist die Anzahl der Optionen, die ich habe. Ich kann z.B. mich mirgense entscheiden, ob ich Die S-Bahn, das Auto oder das Fahrrad nehme. Diese Flexibilität habe ich durch eine bewusste Entscheidung des Wohnorts erlangt, zahle dafür aber auch einen Preis.



**Ingo Eichhorst** ✓ • 1st

AI Architect | CTO | Tech Trainer at IONOS & Universities

5d ...

**Ralf D. Müller** Es gibt ja die gute Idee, dass man Entscheidungen so spät wie möglich treffen sollte. Vor der Entscheidung haben wir also Flexibilität und nach der Entscheidung keine mehr. Das Treffen der Entscheidung ist Architekturarbeit. Wir schränken Flexibilität ein. ...more

Like · 🗨️ 1 | Reply



**Ralf D. Müller** • 1st

Claude.ai Operator :-) @ DB Systel GmbH | iSAQB® CPSA®-A Certified...

5d ...

valide Sichtweise :-)



**Michael Meierhoff**  • 2nd

5d ...

Enterprise Architect at Aviv

Ich wähle ‚Anderes‘: Flexibilität ist immer kontextabhängig – was in einem Atomkraftwerk gilt, unterscheidet sich grundlegend von einer Internet-Plattform. Am Ende misst sie sich an den Change-Kosten und entsteht erst im Zusammenspiel von Technik, Fachlichkeit und Organisation.

#### Beispiel 1: Flugzeug vs. Online-Shop

In der Luftfahrt-Software bedeutet Flexibilität, dass Systeme auch nach Änderungen unter allen Umständen stabil und zertifizierbar bleiben müssen. Im Online-Shop dagegen heißt Flexibilität, schnell neue Bezahlmethoden oder Features einzuführen, um den Marktanforderungen zu entsprechen.

#### Beispiel 2: Bankensystem vs. Social-Media-Plattform

In einem Bankensystem bedeutet Flexibilität, regulatorische Änderungen zuverlässig und ohne Risiko umzusetzen. Auf einer Social-Media-Plattform dagegen heißt Flexibilität, ständig neue Interaktionsformen oder Features auszuprobieren.



Sozial



**Sylvia Ensslen** • 3rd+

4d ...

Leiterin Entwicklung Crealis bei ORISA Software GmbH

Ein flexibles Mindset der Entwickelnden. In einem lebenden Softwaresystem sollte man nie irgendwas als "fertig" begreifen. Nichts ist in Stein gemeisselt. Man muss merken, wenn eine vorhandene Architektur nicht mehr passt und dann auch denn Mut haben, die Dinge zu ändern.

Sicher gibt es Basics, die man von Beginn an beachten sollte. Ich bezweifel allerdings, dass man oft in der Lage ist eine Architektur zu erstellen (und umzusetzen), in der alle zukünftigen Anforderungen leicht integrierbar sind. Der Wille zur Veränderung ist deshalb für mich das Wichtigste.



**Jan Moser** ✓ • 1st

4d ...

Tech Visionary, Thought Leader & DJ 💙 |Opinionated and Tattooed 📍 | Brin...

Mindset....alle obigen Massnahmen helfen dabei flexibel zu sein...wenn aber das Team dies nicht ultimativ will oder versteht (verteilte Systeme sind komplexer als mono- oder modulithische) (und mit Team meine ich Architektur- sowie Development-Rollen) dann bringt dir alle Liebesmüh nichts...



**Micha Pietsch**  • 2nd

6d ...

Senior Software Developer for full stack web applications – 10+ yrs in web de...

Anderes: alle Entwickler im Team fortlaufend dafür sensibilisieren, konkrete Beispiele besprechen, Ruhe („langsam ist präzise und präzise ist schnell“) etc.



**Peter Fichtner** • 1st

5d ...

Techie. Nerd. (Clean) Coder. Technical Agile Coach. Agilist. FOSS Enthusiast/M...

Leider keine Mehrfachauswahl möglich, neben einer guten fachlichen Architektur ist für mich ein wesentlicher Aspekt die Befähigung der Mitarbeitenden: Ich kann auch eine saubere fachliche Architektur technisch komplett zementieren, so dass ich kaum noch Änderungen am System/der Architektur des Systems vornehmen kann.



**billie85.bsky.social** @billie85.bsky.social · 6d

Zumindest nicht mit Zentralisierung von der ich leider betroffen bin. Ich kann also nur theoretisch davon plaudern, aber lose Kopplung ist das erste, was mir einfällt.



**Daniel Novak** ✓ • 2nd

5d ...

Software Engineer | Spar

Meine Erfahrung zeigt, dass es immer eine 3er Kombination ist die auf Dauer am meisten Erfolg gebracht hat. In den meisten Fällen wurde jedoch auf mindestens einen der Bausteine vergessen:

1. Gute Architektur die zu dem System passt
2. Tools zur Sicherstellung der Requirements der Architektur (keine Architekturverletzungen)
3. Schulung und Verständnis der Prinzipien dahinter.

Meistens fehlte es an Punkt 3, egal ob bei Modulith, Hexagonal, Microservices etc, weshalb Dinge unschön wurden und dann wieder der Plan nicht so aufgegangen ist wie er sollte.

Wenn man neue Kollegen, egal ob Senior oder Junior nicht abholt und schult, bauen sie an den falschen Ecken rum.



**Nicolai Wolko**  • 1st

6d ...

Softwarearchitektur. Publikationen. Lehre. Gutachten.

Da stellt sich mir zunächst die Frage, was „flexibel“ überhaupt bedeutet. Für mich ist eine Architektur dann flexibel, wenn sie klare Verantwortlichkeiten, hohe Kohärenz und geringe Kopplung aufweist.

In der Praxis oft nur sehr schwer erreichbar: Technik und Know-how allein reichen nicht. Es braucht über längere Zeit gelebte Disziplin. Dazu gehören technisches Können, ein Umfeld mit Bewusstsein für Qualität – von der Anforderungsermittlung bis zur Abnahme – und der Wille, nicht dem schnellen Ergebnis hinterherzulaufen.

Das "Wie" ist oft bekannt oder zumindest erahnbar. Entscheidend ist der konsequente Weg dahin. Wenn ich es auf eine Antwort reduzieren müsste, dann wäre meine Antwort: Disziplin!



Nicht abstrahieren, wenig bauen



Achtsam Coden <∞ />

@achtsam@norden.social

🕒 6d

[@ewolff](#) nur Funktionen einbauen, die man auch wirklich braucht oder leicht wieder entfernen kann. Den Code immer wieder vereinfachen.

Der Versuchung widerstehen, Dinge im Voraus möglichst abstrakt zu implementieren, damit man sie später erweitern kann. Da man keine Kristallkugel hat, ist die Wahrscheinlichkeit groß, dass man die falsche Abstraktion baut.

Viel mit dem Code arbeiten. Ein Teil der Architektur ist im Kopf. Wenn man sie vergisst, wird man sich mit Anpassungen schwerer tun.



**nexos**

@nexos

🌐 6d

[@ewolff](#) Man nehme eine nicht flexible Architektur. Bricht sie an all den Stellen, an denen sie nicht flexibel genug war. Bei Bedarf noch ein großer Löffel unnötige Flexibilität an anderen Stellen entfernen. Gut umrühren und nett dekoriert servieren.



**Robert**

@fnordfish@ruby.social

5d

@ewolff „don't try to be clever“

Meiner Erfahrung nach stehen Libs/Tools die tief abstrahieren irgendwann im Weg. Alles was das Leben am Anfang leichter macht, macht es später schwerer.



Patrick Wunderlich

@pawu76

🌐 6d

[@ewolff](#) soviel wie nötig und so wenig wie möglich (sorry für die Meta-Antwort)



Wolfgang Müller (DE:er/EN:he)

@muellerwhh@sueden.social

6d\*

@ewolff Ich fand beim original Paper über das Wasserfallmodell super interessant, dass da drinnen steht: Macht das Ganze zweimal. Einmal um zu lernen, und einmal um's zu machen.

Ich denke, eine sinnvoll Flexible Struktur bekommt man, wenn Leute mit tiefer Kenntnis von Software Engineering und tiefer Kenntnis des Problems zusammenarbeiten.



**pv** @pv353.bsky.social · 6d

Dinge erst dann umsetzen, wenn sie unausweichlich sind.  
Nicht so tun, als hätte man Gewissheit, wie sich die Zukunft entwickelt.



**pv** @pv353.bsky.social · 5d

Und Dinge erst vereinheitlichen, wenn sie wirklich mind. 3x  
vorkommen. Wenn man Optionen benötigt, um die Varianten  
abzubilden, eignet es sich wenig für DRY.



**pv** @pv353.bsky.social · 4d

Schade, dass es hier nicht mehr ernsthafte Antworten gab. Ich hätte  
gerne etwas dazugelernt.



**Apollo**

@apollo@nrw.social

🌐 6d

@ewolff ich denke durch KISS. Lieber kleine spezielle Programme als extreme Monolithen. Aber Microservices müssen es in der Regel auch nicht sein, da kommt dann durchs Netzwerk zu viel Komplexität rein.

Wenn ich so drüber nachdenke wird man die Architektur immer wieder hinterfragen und anpassen müssen, sodass es wechselnde Anforderungen erfüllt.





**raupach**

@raupach@wue.social

🌐 6d\*

@ewolff Garnicht erst anfangen. Flexibel endet immer irgendwann. Software ist irgendwie organisch. Klein, wächst, gedeiht. Viele Idee, wünsche und Features. Dann trägt es Früchte. Man erntet. Und irgendwann sind es alte Reben und es wird zeit zu akzeptieren, dass die Software am Ende ist und man pflanzt erneut.



Aus meiner Erfahrung ist die Frage nach „flexibler Software-Architektur“ so allgemein gestellt kaum zu beantworten.

Wenn man „Flexibilität“ als Anforderung formuliert, führt das oft zu Over-Engineering: Es entstehen Frameworks, die „alles können sollen“, in der Praxis aber oft doch nicht alles abdecken – oder nur mit Einschränkungen, z. B. bei UX, Skalierbarkeit oder Wartbarkeit.

Entscheidend ist, dass der Business-UseCase nie aus dem Blick gerät. Der fachliche Kontext muss die Klammer sein – nicht eine vermeintlich universelle Architektur.

Ich sehe das so:

- Nicht versuchen, die perfekte Architektur für alle (un)bekannten Anforderungen im Voraus zu bauen.
- Modularisierung einsetzen, um Bereiche klar zu trennen, Austauschbarkeit und Wartbarkeit zu erhöhen.
- Klare Ziele und Rahmenbedingungen definieren (z. B. erwartete Nutzerzahlen, SLAs).
- Refactoring als normalen Prozess akzeptieren – und bei größeren Änderungen auch den Austausch von Teilen oder des Gesamtsystems.

Am Ende gilt: Man kann nicht alles vordenken. „Flexibilität“ entsteht durch klaren fachlichen Rahmen, Modularisierung und die Bereitschaft zur laufenden Anpassung.



**Jens Rehsack**  • 2nd

6d ...

Senior System Architect (Embedded/IoT) | Business-Technical Translation | Yo...

Wie flexibel muss es denn sein? Also welche Art von Flexibilität stellst Du Dir vor? Wenn ich alles flexibel mache, was kann ich dann noch konkret erreichen? Was habe ich übersehen?

Vielleicht bin ich zu naiv - aber mein Ansatz ist: K.I.S.S. Und dann Prügel im Review abholen. Dann zurück zu K.I.S.S.



**UrsENZler** ✓ • 1st

4d ...

Software Architekt und Unternehmensentwickler bei Calitime AG Conference...

Und nochmals ein Gedanke (es regnet halt immer wieder Benachrichtigungen rein 😊):

Flexibilität entsteht nicht dadurch, dass alles möglichst abstract, generisch, mit vielen Interfaces und überall mit Strategy Pattern gemacht wird. Das führt eher zu einem unverständlichen System, welches dann nicht mehr verstanden und somit nicht mehr angepasst werden kann.

Flexibilität entsteht durch simplen Code, der einfach angepasst und ausgeliefert werden kann.

Flexibel?



**Simon Martinelli** @martinelli.ch · 6d

Flexibel wofür?



**pv** @pv353.bsky.social · 4d

Flexibilität heißt, dass die Anforderungen vorher nicht bekannt sind.



**Simon Martinelli** @martinelli.ch · 4d

Nun ohne Anforderungen keine Architektur



**pv** @pv353.bsky.social · 4d

Schade. Flexibilität ist meiner Erfahrung nach eigentlich immer gefordert und es hätte ja sein können, dass du allgemein anwendbare Best Practices hast.



**mercuriusb** @mercuriusb.bsky.social · 2d

wenn du keine Anforderungen hast, darfst du dich auch nicht über das Resultat beschweren 🙏





**Urs Enzler** ✓ • 1st

6d ...

Software Architekt und Unternehmensentwickler bei Calitime AG Conference...

Zuerst mal eine Frage zurück:

Flexibilität bezüglich was?

- Ändernde Business-Anforderungen (Evolution des Marktes, Business Model, Marktsegment, ...)
- technischer Aspekte (z.B. Modernisierung, Skalierung, ...)
- Architektur-/Designkonzepte (wir wollen Refaktorisieren können, wenn wir eine bessere/einfachere/mächtigere Lösung finden)

Je nach Antwort sind die Mittel dazu etwas anders.

So grundsätzlich wohl nie falsch sind eine gute Modularisierung (entlang Business Capabilities) und Entkopplung von Infrastruktur/Libraries/Frameworks etwa wie in Ports&Adapters-Architektur. Wenn diese beiden Konzepte anwesend sind, dann lässt sich die Software zu allem gewünschten umbauen.





**Stefan Rother-Stübs**

@rotherstuebs@foojay.social

🌙 5d

@ewolff

Das hängt davon ab, was in dem Fall mit "Flexibel" gemeint und erreicht werden soll

# Modularisierung



**Martin Axiomer**

@axiomer@23.social

@ewolff Modularisierung

🌙 6d



**billie85.bsky.social** @billie85.bsky.social · 6d

Flexibel bedeutet ja, dass es relativ leicht an neue Anforderungen angepasst werden kann. Das erreicht man nicht, wenn Teile verschachtelt und gegenseitig voneinander abhängig sind. Das erreicht man eher, wenn man einzelne Teile wiederverwenden, neu einfügen und neu anordnen kann.



**Thomas Schaper**  • 2nd

6d ...

"The only thing which will redeem mankind is cooperation." (Bertrand Russell)

Saubere Trennung des fachlichen Kerns von technischen Aspekten und Schnittstellen. Diese Zielsetzung wollte man sogar schon mit den ersten EJB-Versionen Ende der 90er erreichen... Die hexagonale Architektur soll dies nun endlich ermöglichen.



**Andreas Koop**  • 2nd

6d ...

Digitalisierung gemeinsam meistern - Softwarearchitekt und KI Enabler für di...

Teile die Anwendung in unabhängige, lose gekoppelte Module auf.





**Stefan Armbruster** @armbruster-it.de · 4d

Wie die Frage, so die Antwort: it depends.





**Tammo van Lessen** @vanto.bsky.social · 6d

Das weißt du doch selbst... 🕶️



**Roland Weisleder** @rweisleder.de · 6d

Map<Object, Object> data;

Alles



**Kim Nielsen**  • 1st

CTO @ Vitec Cito (Vitec Software Group)

Alle 3!

6d ...



**Alexander von Zitzewitz**  • 1st

(edited) 6d ...

Helping software architects to achieve better results by implementin...

Ich hätte gerne alle 3 selektiert, denn Technik (Tools), organisatorische Massnahmen und die fachliche Architektur gehören zusammen. Die schönste Architektur ist nutzlos, wenn der Code sich nicht daran hält.



**Patrick Müller** • 1st

3d ...

Wenn Du ein Schiff bauen willst, dann trommle nicht Männer zusammen, um ...

Ich sehe einen guten Mix aus allen Maßnahmen Eberhard. Nur mit einer gut verstandenen Fachlichkeit, Bebauung, Maßnahmensteuerung, Architektur und Technik ... UND den organisatorischen Rahmenbedingungen, kann Flexibilität erreicht und vor allem im Unternehmen gelebt werden



**Guido Altmann**  • 2nd

5d ...

 Fullstack .NET Developer | C#, Blazor, REST, SQL | Effizient. Architekturbew...

Mindestens alle drei Punkte sind wichtig. Und das ganze Team muss an einem Strang ziehen.



**Simon Meurer** ✓ • 2nd

Head of Software Development (People) at INVERS

5d ...

Irgendwie alle 3 Dinge und vor allem herausfinden was im entsprechenden Fall flexibel heißt 😊





**Ralf Bamert**  • 2nd

4d ...

Making Digital Useful - Innovation by software, data and networks.

Flexibilität entsteht nicht durch eine Maßnahme, sondern durch das Zusammenspiel. Technik schafft die Basis (APIs, lose Kopplung, Automatisierung), die fachliche Architektur sorgt für Stabilität im Modell, und organisatorische Maßnahmen geben Teams die Freiheit, Entscheidungen nah am Problem zu treffen. Wer nur auf einen Aspekt setzt, baut früher oder später ein neues Legacy-System.



**Erwin Glassée** ✓ • 2nd

(edited) 6d ...

Enterprise Architect at Myreas (Colruyt Group)

Alle oben genannten. Aber zuerst fachlich. Wenn das nicht stimmt, ist alle Flexibilität verloren. Mit Microservices ist es dann eben noch etwas schlimmer wie mit Monolith.

Wenn es fachlich stimmt, hat man ohne Technik oder Organisation noch eine kleine Chance.



**Christian Schmidt** ✓ • 2nd

5d ...

Papa & IT Consultant & Software Architect bei Qvest Digital AG

Ist für mich immer ein Dreiklang aus Technik, Fachdomäne und Entwicklungsorganisation.

Wenn eins davon ein monolithischer Klumpatsch ist, kämpft man immer dagegen an. Ist die Fachdomäne unstrukturiert, kann sich eine Architektur schlecht daran orientieren oder muss dagegen arbeiten.

Ist der Entwicklungsprozess unstrukturiert, greift hier gerne Conways Law und auch hier kämpft man dagegen an.

Ist die Technik unstrukturiert hat man es dagegen (als "Techniker") noch am Einfachsten, weil man hier tendenziell mehr Hebel hat

Aber ich denke, man muss alle drei Bereiche ganzheitlich betrachten und alle Ebenen versuchen mit zu nehmen.

Fachlichkeit



**Jan Wedel**

@stealthmusic@mastodon.online

🌐 4d

@ewolff ist das nicht schon ein Widerspruch in sich? :) jedenfalls sagt man ja, dass Architektur das an der SW ist, was am schwersten / teuersten zu ändern ist.

Da ich schon einmal ein Produkt in eine neue Architektur „pressen“ bzw. „zerschneiden“ musste, würde ich sagen: Modularisierung nach Domänen und eine gewisse Abstraktion der Äußeren Schnittstellen und rein technischem Code hatte geholfen.



**Jan Wedel**

@stealthmusic@mastodon.online

🌐 4d

@ewolff Ansonsten würde ich fast behaupten, das „flexibel“ im Normalfall als Qualitätskriterium für Architektur nicht unbedingt erstrebenswert ist. Außer es ist mit „flexibel“ gemeint, dass ich die SW innerhalb der gewählten Architektur leicht anpassen (erweiterbar, portierbar etc.) kann.



**Axel Rotter Morgado** ✓ • 1st

Software Architect at GRENKE AG

5d ...

Die fehlende Fachlichkeit ist normalerweise die größte schwachstelle in meiner Erfahrung.



**Michael Trapp** ✓ • 1st

5d ...

Freelance Software Developer and Architect - Dedicated to Software

Zunächst sollte man sich mit dem Business beschäftigen und sich die Zeit nehmen, um zu verstehen, worum es geht und wo das System wirklich flexibel sein muss. Ein System in Bereichen flexibel zu gestalten, die nicht flexibel sein müssen, ist verschwendete Energie, Zeit und Geld. Flexibel zu sein, wo es darauf ankommt, ist ein echter Wettbewerbsvorteil.





**Anatoli Lichii** ✓ • 1st

4d ...

Mit maßgeschneiderten IT-Lösungen und modernster Technologie in die Zuk...

- Verständnis von Business Needs
- Commitment auf der Management Ebene für nachhaltige Entscheidungen

Dann kann man die Architektur wirklich anfangen flexibel zu gestalten.  
Ansonsten führt es zum Spagetti.



**Martin Dilger**  • 1st

(edited) 6d ...

So verwandeln Software-Teams chaotische Projekte in planbare Proz...

Kollaborative Software-Modellierung ( das Ergebnis davon ist die fachliche Architektur ) und Abbildung der Fachlichkeit in kleinen fachlich geschnittenen Modulen ( Slices )

Technik alleine hilft nicht - und für die saubere Umsetzung muss die Organisation bereit sein.

Es gibt nicht den einen Punkt, der eine gute Architektur ermöglicht - im Zweifel aber ist kollaborative Modellierung der stärkste Enabler - denn nichts bringt Architekturen schneller zu Fall als falsche Annahmen und falsch verstandene Anforderungen.



**Martin Pasch** ✓ • 2nd

6d ...

VP Advanced Products&Services bei Voith Hydro

Die fachliche Seite ist heutzutage Gegenstand ständiger Anpassungen. Somit kann ich nur flexibel sein, wenn ich die fachliche Seite mit guter Technik kombiniere. In komplexen Unternehmensstrukturen kann z.B. eine rein hierarchische Strukturierung nie alle use-cases erfüllen, ein Graph muss her. Microservices im klassischen Sinne benötigen ein 'right-sizing', Datenanpassungen sollten schnell im UI sichtbar werden, ohne dass ein UX designer in Ohnmacht fällt, daher kommt die Code-Generierung hinzu...

Anderes



**Christian Gudrian**

@cgudrian@social.tchncs.de

🌐 6d\*

@ewolff Schreibe keinen Code, der sich leicht wiederverwenden lässt;  
schreibe Code, der sich leicht wegwerfen lässt.



**Apollo**

@apollo@nrw.social

🌐 6d

@cgudrian @ewolff uh, das gefällt mir.



 **Raf Lefever**  • 2nd  
Sw.Arch. | Erlang/Elixir, .NET

(edited) 5d ...

Architecture, nor the processes you follow, nor the way an organisation is set up, will produce a well-oiled (software) factory on their own. They form a triad, which I like to call 'The Triad of Excellence'





**Urs Enzler** ✓ • 1st

5d ...

Software Architekt und Unternehmensentwickler bei Calitime AG Conference...

Und noch ein Gedanke:

Ich denke Architektur nicht in fixen Zuständen, sondern in Evolutionspfaden. Denn die Architektur wird sich meist entwickeln müssen und wir können auch oft schon etwa abschätzen, was evtl. möglich sein sollte. Als Beispiel: Unsere Applikation ist mit 100 Benutzern gestartet und wir haben das (ökonomische) Ziel von 25000 Benutzern. So konnten wir mögliche Pfade entwerfen (verschiedene Abfolgen von Optionen zu Architekturentscheidungen), wie wir die Architektur zu Beginn nicht überdimensionieren und so schneller am Markt sind und uns aber keine Sackgassen einfangen, wenn wir die Architektur weiterentwickeln.

Dies haben wir für die für uns wichtigen Architekturaspekte und Qualitätseigenschaften gemacht. Dies hilft zu erkennen, wo Flexibilität wichtig ist. Aber Überraschungen gibt's halt immer noch 😊 .



**Jet Loong Tang** ✓ • 3rd+

Softwareersteller

(edited) 5d ...

**Urs Enzler**, ich teile den Gedanken. Ich habe meist dann einen Wardley Map gedanklich vor mir. Es gibt mir Ruhe anzuhalten, weil der aktuelle Zustand des Code Managements "just good enough" ist. Und es beruhigt die Spannung zw. Status jetzt und Wunsch/Erwartung zu verstehen oder zu kommunizieren, weil wir einfach noch nicht in ...more

Like | Reply



**Urs Enzler** ✓ • 1st

5d ...

Software Architekt und Unternehmensentwickler bei Calitime AG Confe...

**Jet Loong Tang** kannst du mir deinen Zusammenhang mit Wardley Maps erklären? Wenn ich an unsere Pfade denke, dann sind das selten Verschiebungen in der Wardley Map, sondern andere Lösungsoptionen am selben Ort in der Karte.

Like | Reply



**Jet Loong Tang** ✓ • 3rd+

Softwareersteller

(edited) 4d ...

**Urs Enzler**. mir kam das Gedankenmodell Wardley Map und der Vortrag von Markus Harrer "Evolutionäre Softwarequalität" in den Sinn, als du geschrieben hast, dass der Zustand einer Architektur mit der Zeit (bzw. Nachfrage/Wünschen/Needs) sich meist weiter entwickeln muss. Das Modell veranschaulichte dann, dass in den Phasen manche ...more

# Residuality Theory





**UrsENZler** ✓ • 1st

6d ...

Software Architekt und Unternehmensentwickler bei Calitime AG Conference...

Barry O'Reilly würde hier wahrscheinlich einwerfen, dass es keine Flexibilität braucht, sondern eine resiliente Architektur mittels Residuality Theory. Aber auch nach mehreren Videos und dem Lesen seines Buchs bin ich dazu immer noch nicht richtig schlau.