

Beheersbaarheid software en architectuur



Adviezen voor betere software

- Gebruik saaie technologie – voorkomt overdrachtsproblemen
- Twee keer meten – 1x zagen : bespreek belangrijke keuzes aan begin
- devOps voorkomt problemen bij overdracht aan beheer
- Scrum voorkomt verrassingen
- Goede architectuur helpt met het beheersbaar maken van grotere systemen

Managementsamenvatting architectuur

- Architectuur is nog steeds relevant voor software-ontwikkeling
 - Bij kleine systemen kun je vaak kiezen voor een eenvoudige architectuur volgens één bekend principe
 - Bij grotere systemen moeten meerdere principes op de juiste manier worden toegepast
- In deze presentatie geven we een overzicht van principes
- Voor een aantal principes geven we ook uitleg wat een juiste toepassing inhoudt:
 - Architectuur volgens lagenprincipe
 - Componentgebaseerde architectuur

What Is an Architectural Style?

- An architectural style, sometimes called an architectural pattern, is a set of principles— a coarse grained pattern that provides an abstract framework for a family of systems.
- An architectural style improves partitioning and promotes design reuse by providing solutions to frequently recurring problems.

Categories of styles

| Category Architecture styles | Category Architecture styles |
|------------------------------|--|
| Communication | Service-Oriented Architecture (SOA), Message Bus |
| Deployment | Client/Server, N-Tier, 3-Tier |
| Domain | Domain Driven Design |
| Structure | Component-Based, Object-Oriented, Layered Architecture |

J.D. Meier, David Hill et. al (2009) *Microsoft Application Architecture Guide*, 2nd Edition. Microsoft Press, 2009. Chapter 3: Architectural Patterns and Styles. <http://msdn.microsoft.com/en-us/library/ee658117.aspx>

Tenslotte: meten is weten

- Er zijn technieken en ook tools voor het meten van code-aspecten
- Deze worden niet altijd gebruikt bij acceptatie – gemiste kans om kwaliteitsbesef te verhogen

Appendix

Summary of Key Architectural Styles

| Architecture style | Description |
|-------------------------------------|--|
| Client/Server | Segregates the system into two applications, where the client makes requests to the server. In many cases, the server is a database with application logic represented as stored procedures. |
| Component-Based Architecture | Decomposes application design into reusable functional or logical components that expose well-defined communication interfaces. |
| Domain Driven Design | An object-oriented architectural style focused on modeling a business domain and defining business objects based on entities within the business domain. |
| Layered Architecture | Partitions the concerns of the application into stacked groups (layers). |

J.D. Meier, David Hill et. al (2009) *Microsoft Application Architecture Guide*, 2nd Edition. Microsoft Press, 2009.
Chapter 3: Architectural Patterns and Styles. <http://msdn.microsoft.com/en-us/library/ee658117.aspx>

Summary of Key Architectural Styles

| Architecture style | Description |
|--|--|
| Message Bus | An architecture style that prescribes use of a software system that can receive and send messages using one or more communication channels, so that applications can interact without needing to know specific details about each other. |
| N-Tier / 3-Tier | Segregates functionality into separate segments in much the same way as the layered style, but with each segment being a tier located on a physically separate computer. |
| Object-Oriented | A design paradigm based on division of responsibilities for an application or system into individual reusable and self-sufficient objects, each containing the data and the behavior relevant to the object. |
| Service-Oriented Architecture (SOA) | Refers to applications that expose and consume functionality as a service using contracts and messages. |

J.D. Meier, David Hill et. al (2009) *Microsoft Application Architecture Guide*, 2nd Edition. Microsoft Press, 2009.
Chapter 3: Architectural Patterns and Styles. <http://msdn.microsoft.com/en-us/library/ee658117.aspx>