



arTchitecture

TS8226 - The Art of Designing a Software Architecture

Speaker: Arnoud de Kuijper
CEO T&M Solutions - Netherlands



Redefining software architecting

‘To do for software architecture,
what design patterns did for programming’



CONTENT

- Speaker bio
- Software Architecture?!
- Competences for a good architect
- Using the Design Kaleidoscope
- Summary

Speak bio

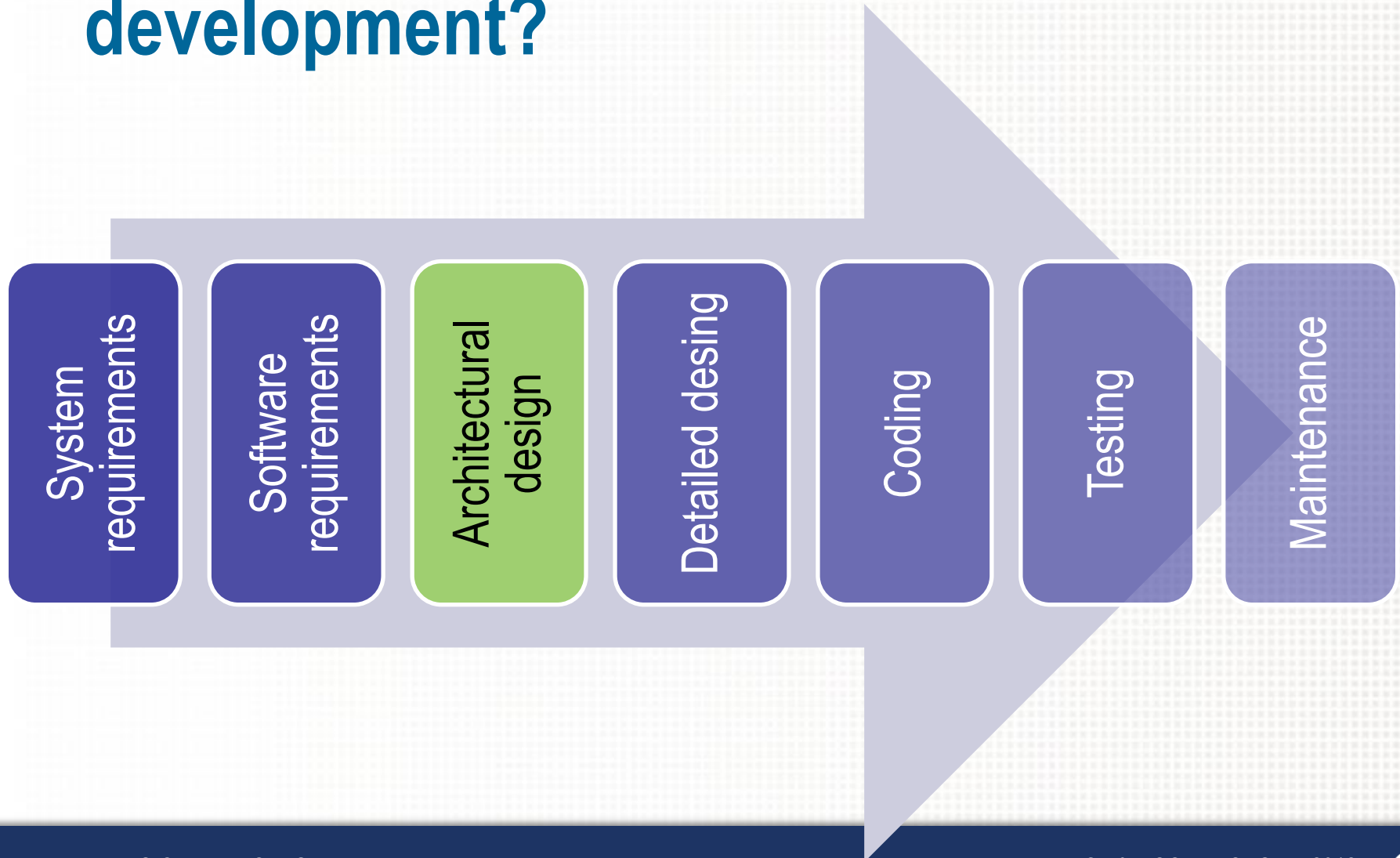
- 15+ years experience in software engineering
- 9+ years LabVIEW
- LabVIEW Architect and Certified Prof. Instructor
- Mechanical, Biomedical and Business Administration
- Hobbies: Software and electronics

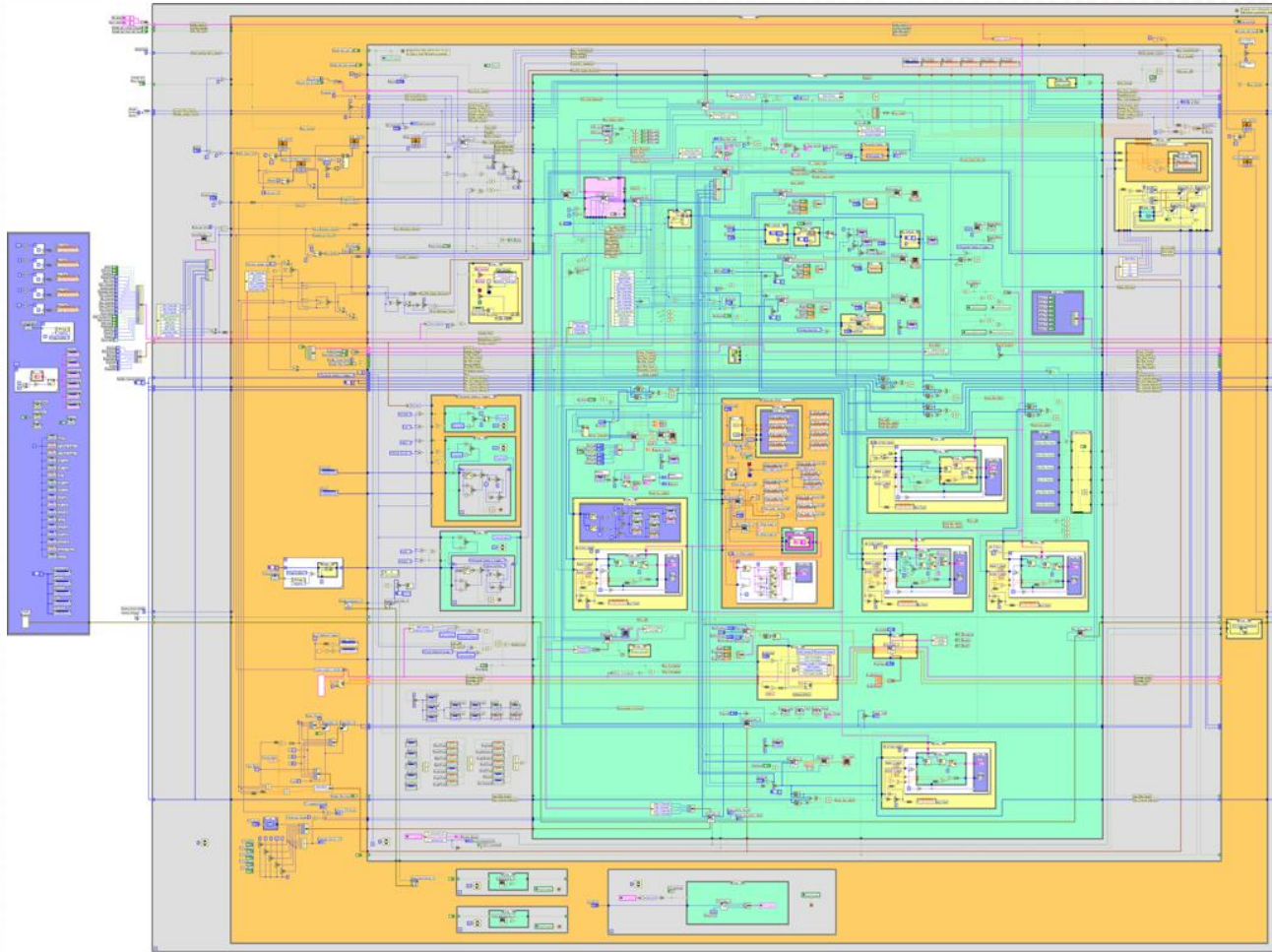


What is a software architecture?

- SA can be defined as a **Abstraction, overarching design**
- that '*describes the* **Functionality and protection** **Tasks and responsibilities** *and*
- *of a software-system and the*
- **Dependency** *between the components*'.

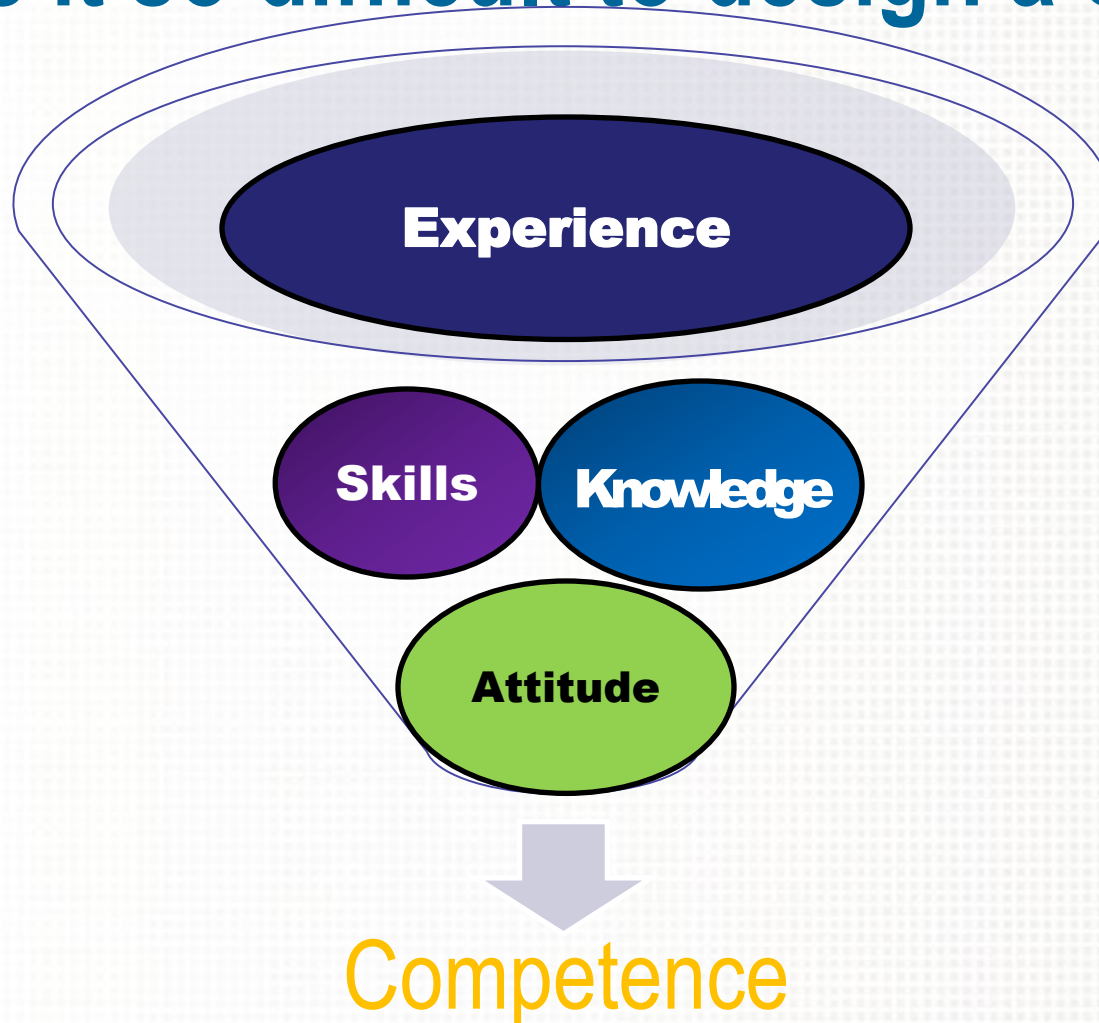
Where does architecture fit in development?





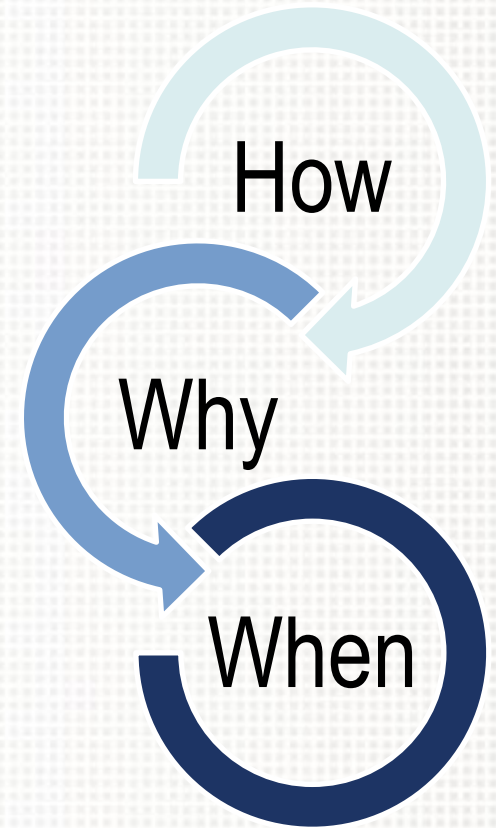
Why should we care?

Why is it so difficult to design a SA?



Competent SA developer

- **Skills** (How to use building blocks)
- **Knowledge** (Understanding)
- **Attitude** (Preference and principle)
- **Experience** (Weigh your options)



Skills

- Object Oriented vs Functional Oriented
- Sub-systems in architecture
- Interloop dependency

OOD vs FOD

OO Design

- Think in 'objects' as scope
- *The object instrument contains all belonging 'functions'.*
- Object is a 'noun'
- Object has methods
- Object can store data
- Objects have relations
- Objects have hierarchy

FO Design

- Think in 'functions' executed as a task.
- *The function measurement connects to an instrument and takes a sample.*
- Task oriented (verbs)

OO in My opinion

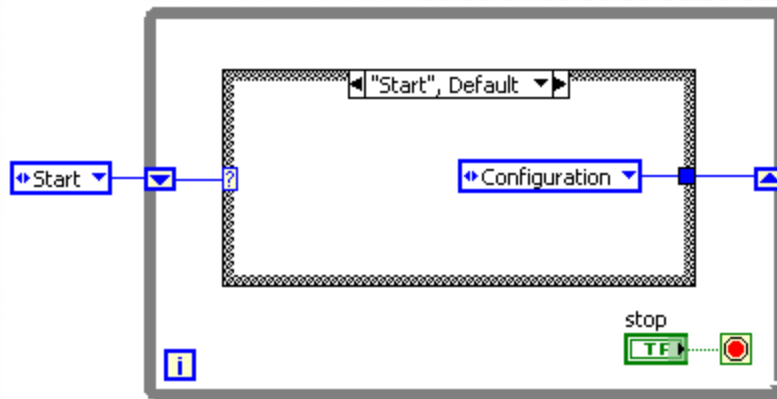
Pro's

- Re-use of code
- Support in DEV environment for:
 - Override
 - Protection
 - Hierarchy
- Standardized design patterns

Con's

- Additional abstraction layer
- Difficult to define the proper class
- By reference or by value?
- Mix with OOD and FOD can confuse.
- FOD has a big legacy

Atom of engineering: Statemachine



The statemachine concept; It is always applicable!

Benefits

- Sequencing
- Repeat/retry
- Data transfer between states
- Abort conditions
- Scoping

Interloop relationship options

Timing & data transfer

- Fire and forget (notify)
- Send and receive [with acknowledge] (queue)
- Broadcast (notify or event)
- Poll (variables)
- ...

Does your requirement fit your mechanism..?

Little quiz.....

- I have 2 while loops with each different execution times for analysis. How do I make sure they execute the same number of iterations?

Possible answers:

Timers

Data relationship

Rendevouz

....

Little quiz.....

- I have 1 system that needs to share it's data with several dynamically spawned systems.

Possible answers:

Queue's (Queue is for 1 to 1 relationship with buffer)

Notifiers (Have multiple listeners, but buffer=1)

Event's (Multiple systems can register and has a buffer per listener)

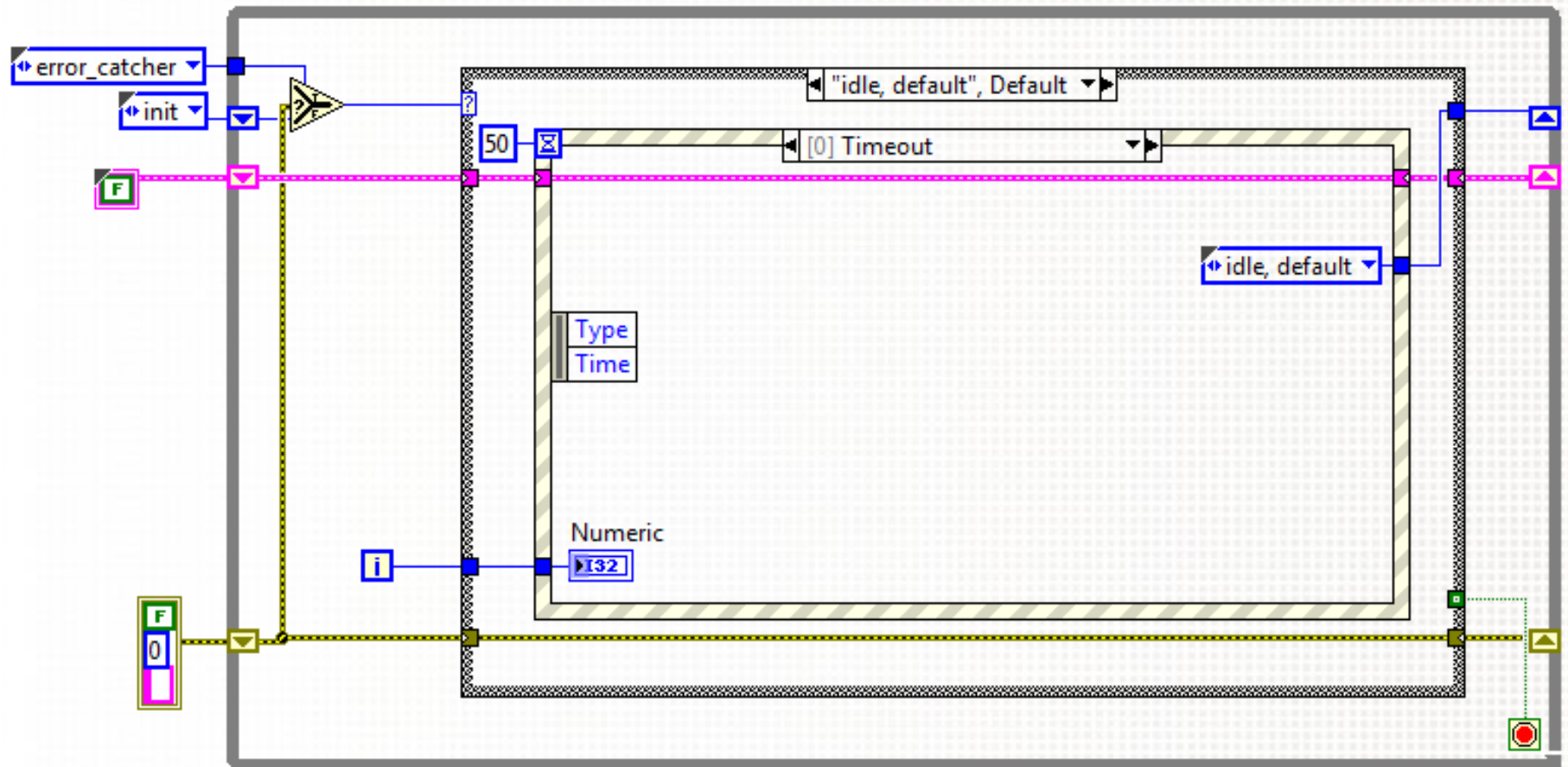
....

Knowledge

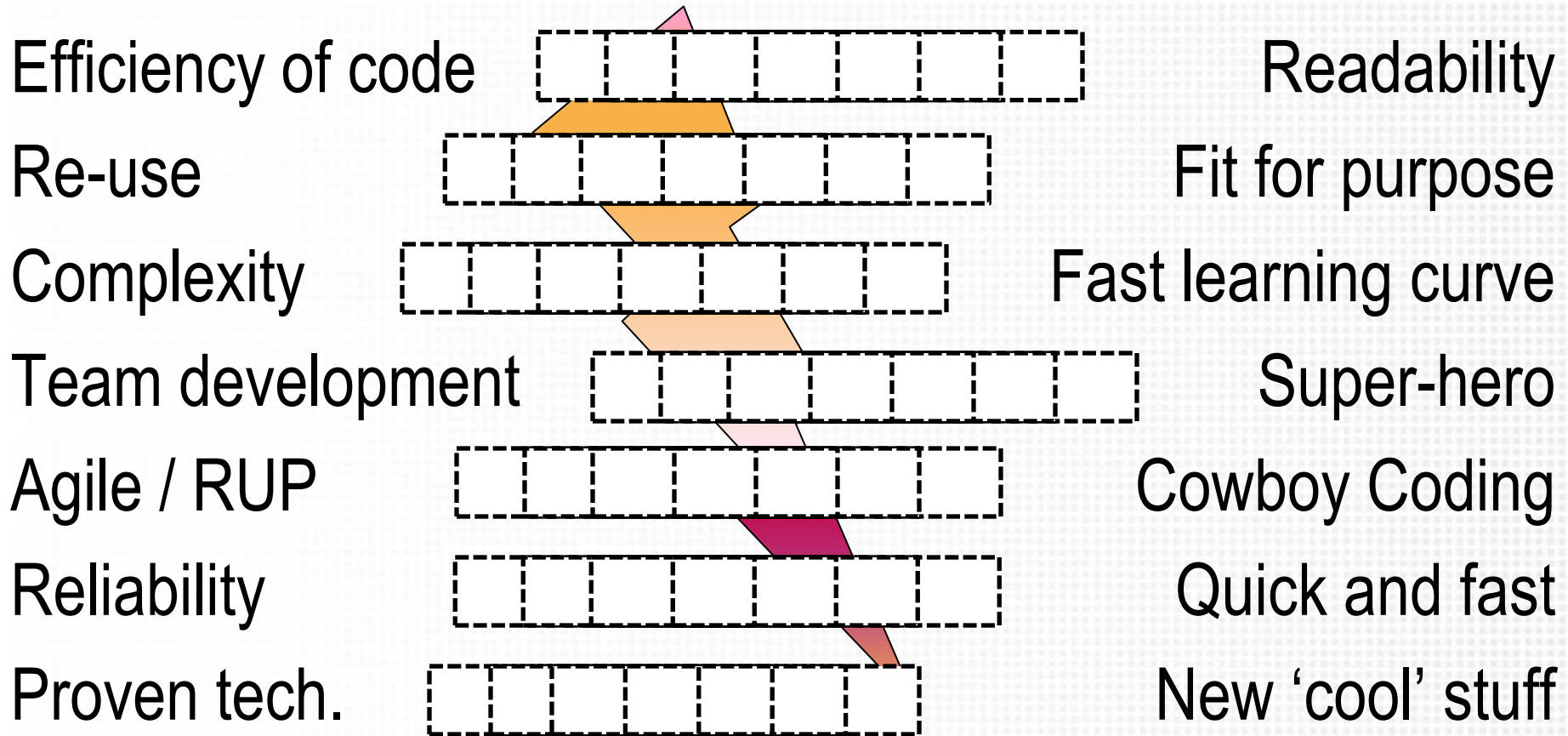
- Learn design pattern applicability by heart
 - Statemachine
 - Product / consumer
 - Deamon, etc.
- Know all synchronization options and their behavior (when to use a queue, semaphore, etc)
- Question every behavior you cannot explain..



Knowledge allows you to leverage

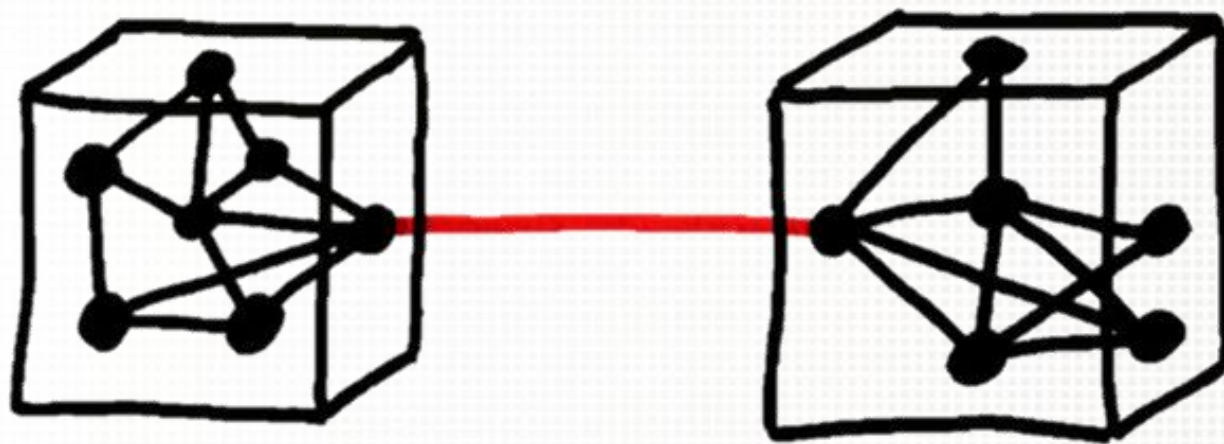


Attitude – Preferences & Principles



Experience

COUPLING AND COHESION

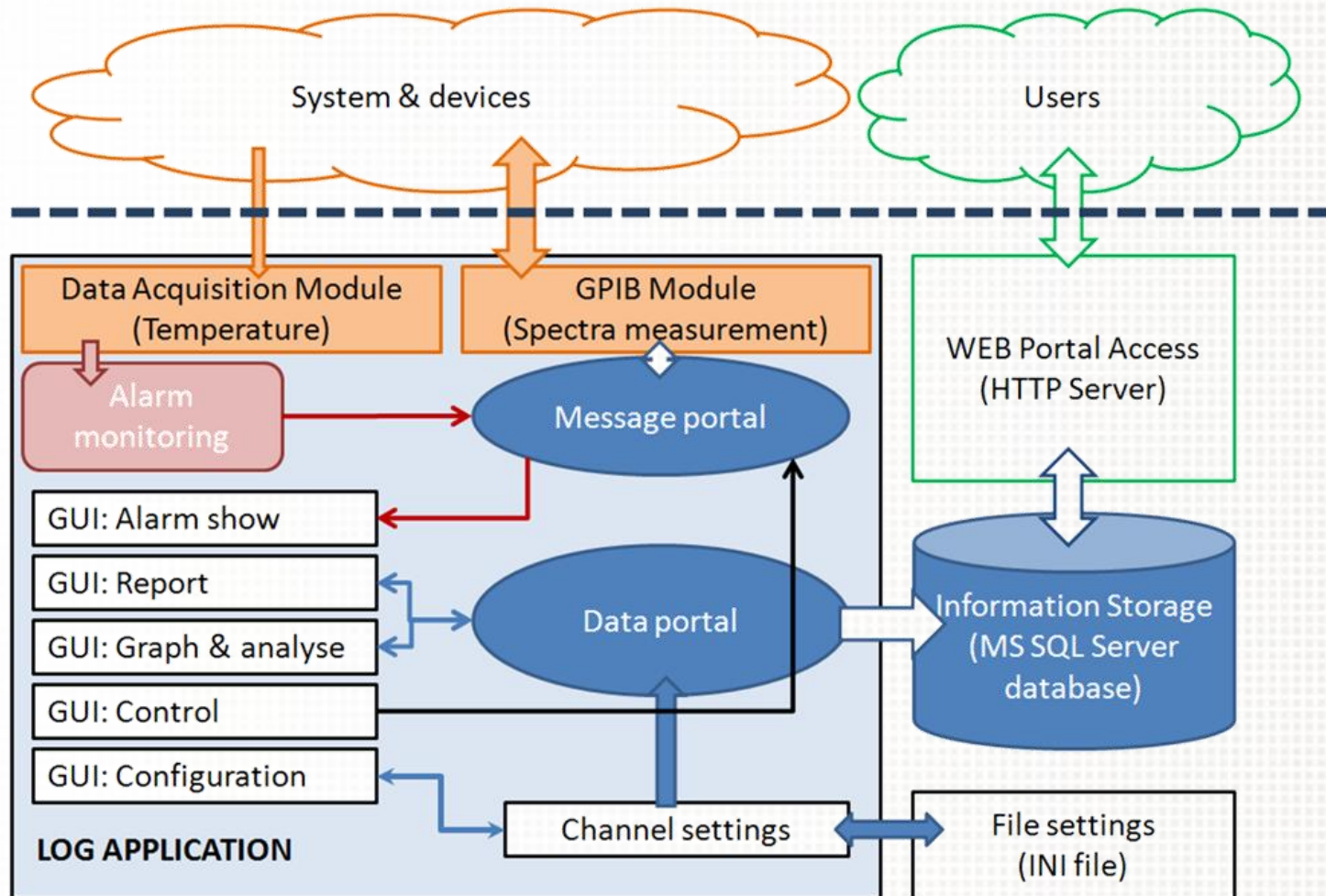


<http://www.planetgeek.ch/2011/07/08/presentation-agile-code-design-how-to-keep-your-code-flexible/>

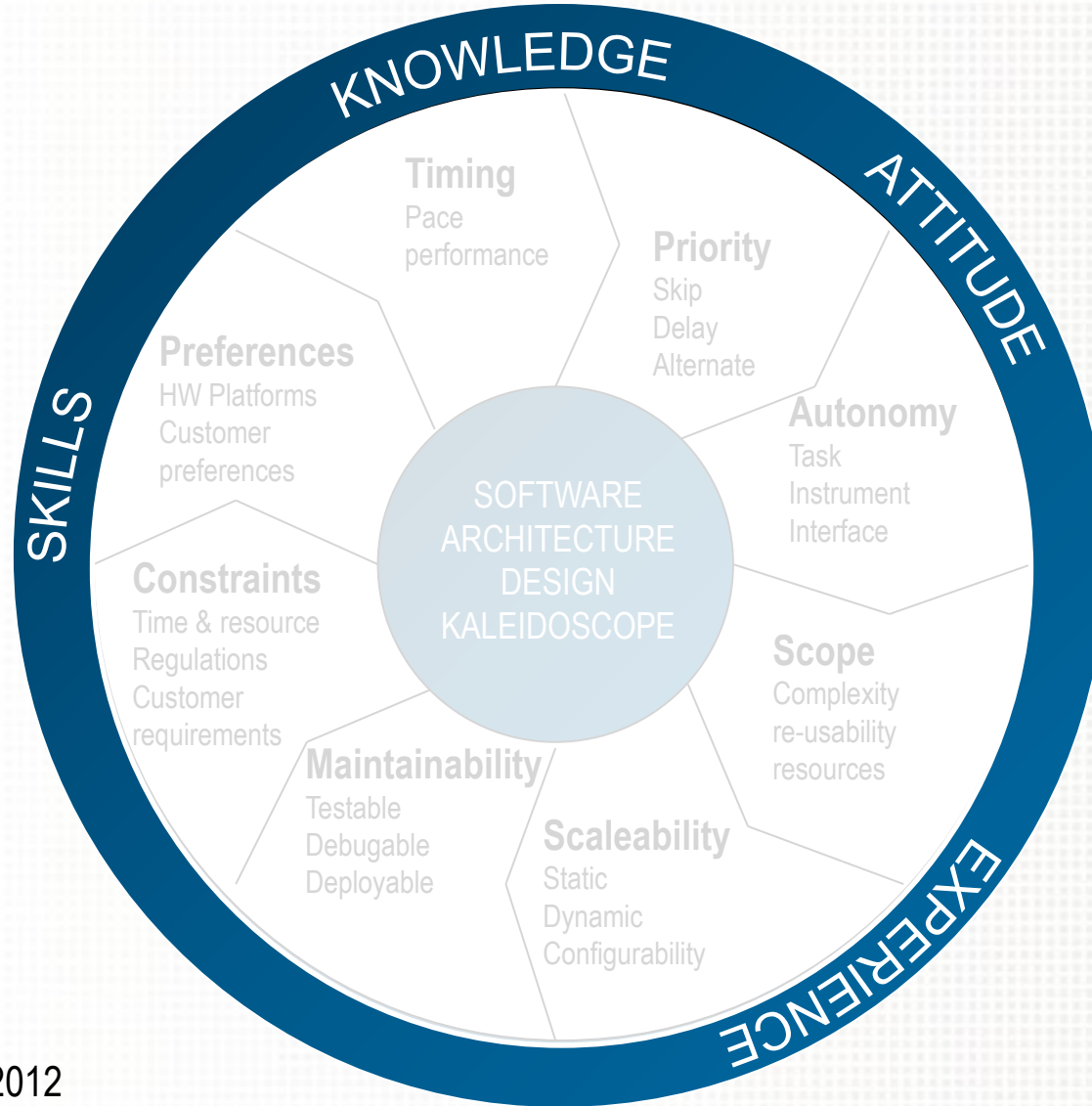
2 important question to ask!

- What is the most appropriate partitioning into sub-systems. (Design kaleidoscope)
- How do I maintain relationships and by what mechanism. (Interloop communication)

Possible outcome of SA



DESIGN KALEIDOSCOPE

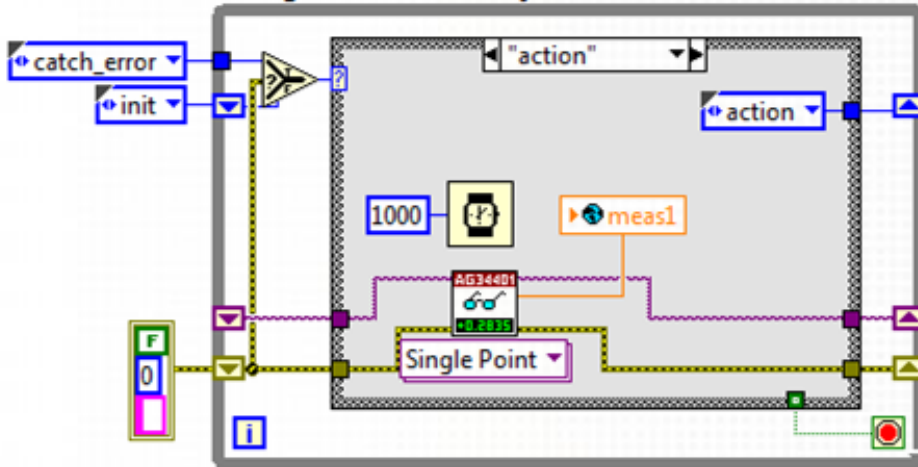


© Arnoud de Kuijper, 2012

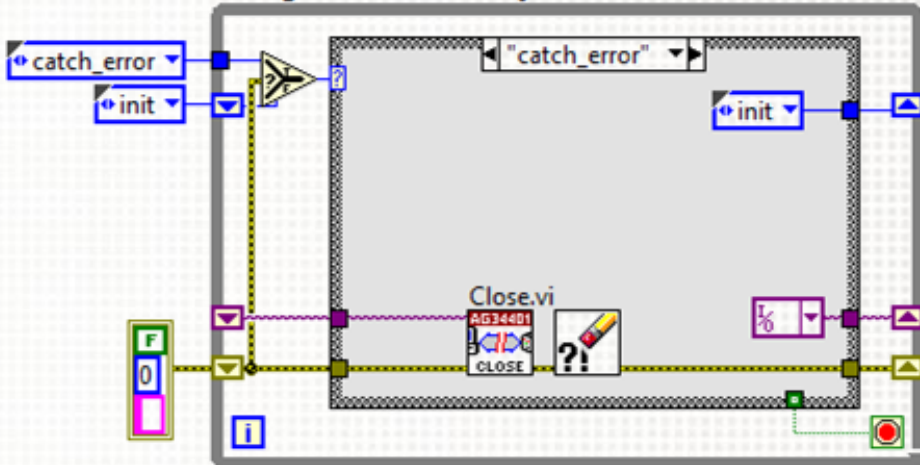


Device based partitioning

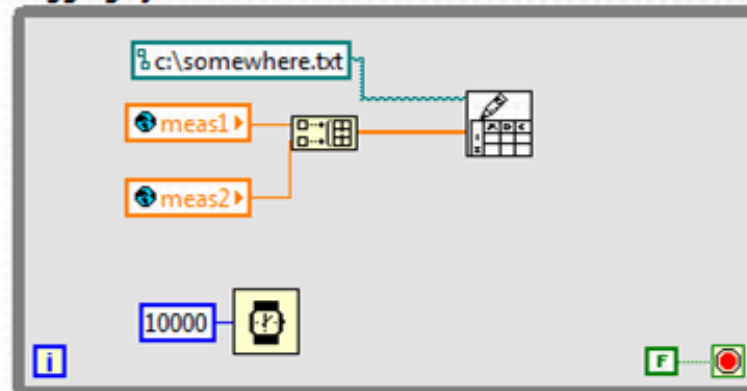
Voltage Measurement System 1



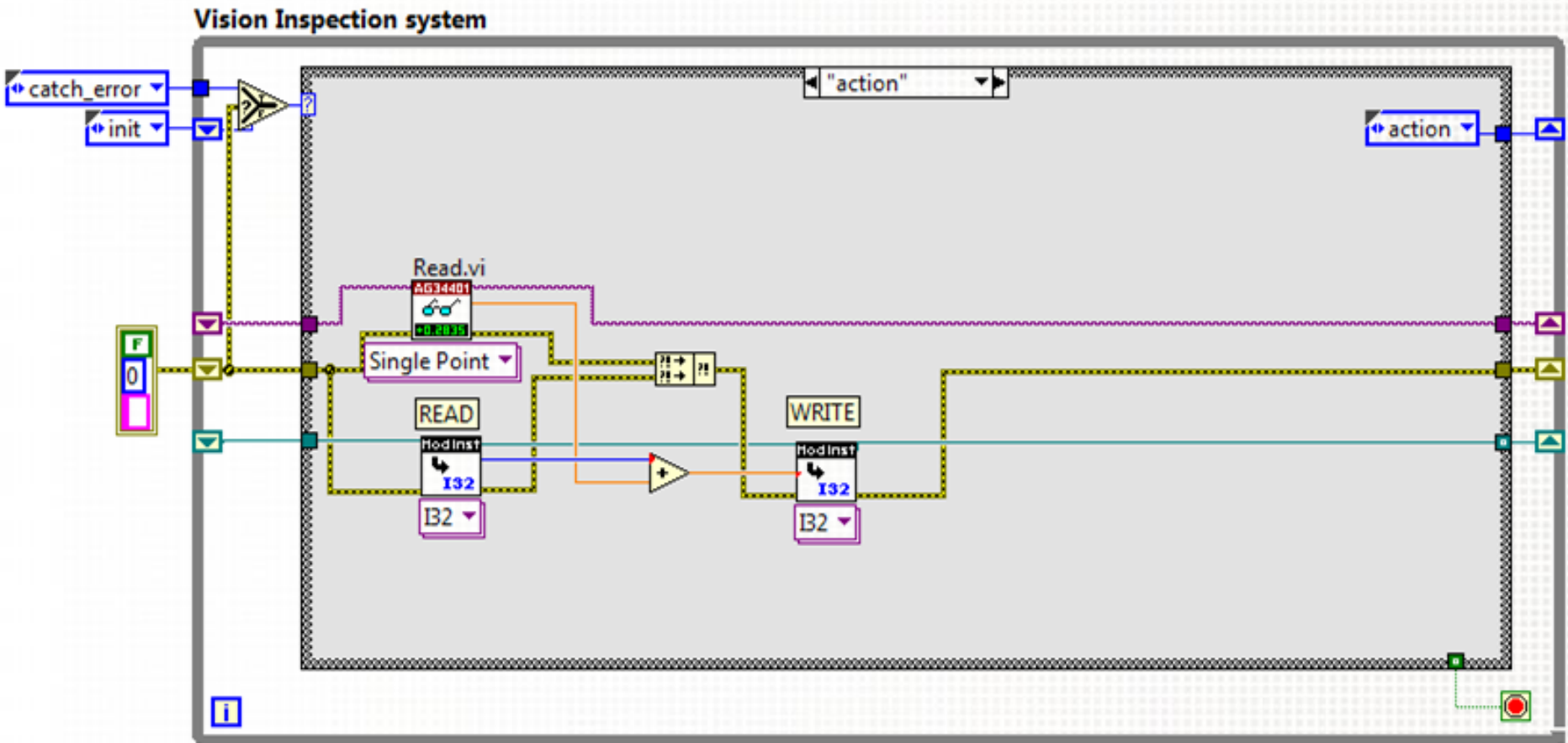
Voltage Measurement System 2



Logging system status



Task based



Summary

- Improve skills by training courses
- Improve knowledge by questioning
- Develop attitude by peers and context
- Gain experience by Re-Thinking

-> Abstraction reduces complexity

Stay in contact

- *www.tm-solutions.nl*
- *www.esi-cit.com*
- *nl.linkedin.com/in/adekuijper*



Stay **Connected** During and After NIWeek



ni.com/niweekcommunity



facebook.com/NIWEEK



twitter.com/#!/niweek



<http://linkd.in/ljfwyB>



youtube.com/niglobal



linkd.in/adekuijper



