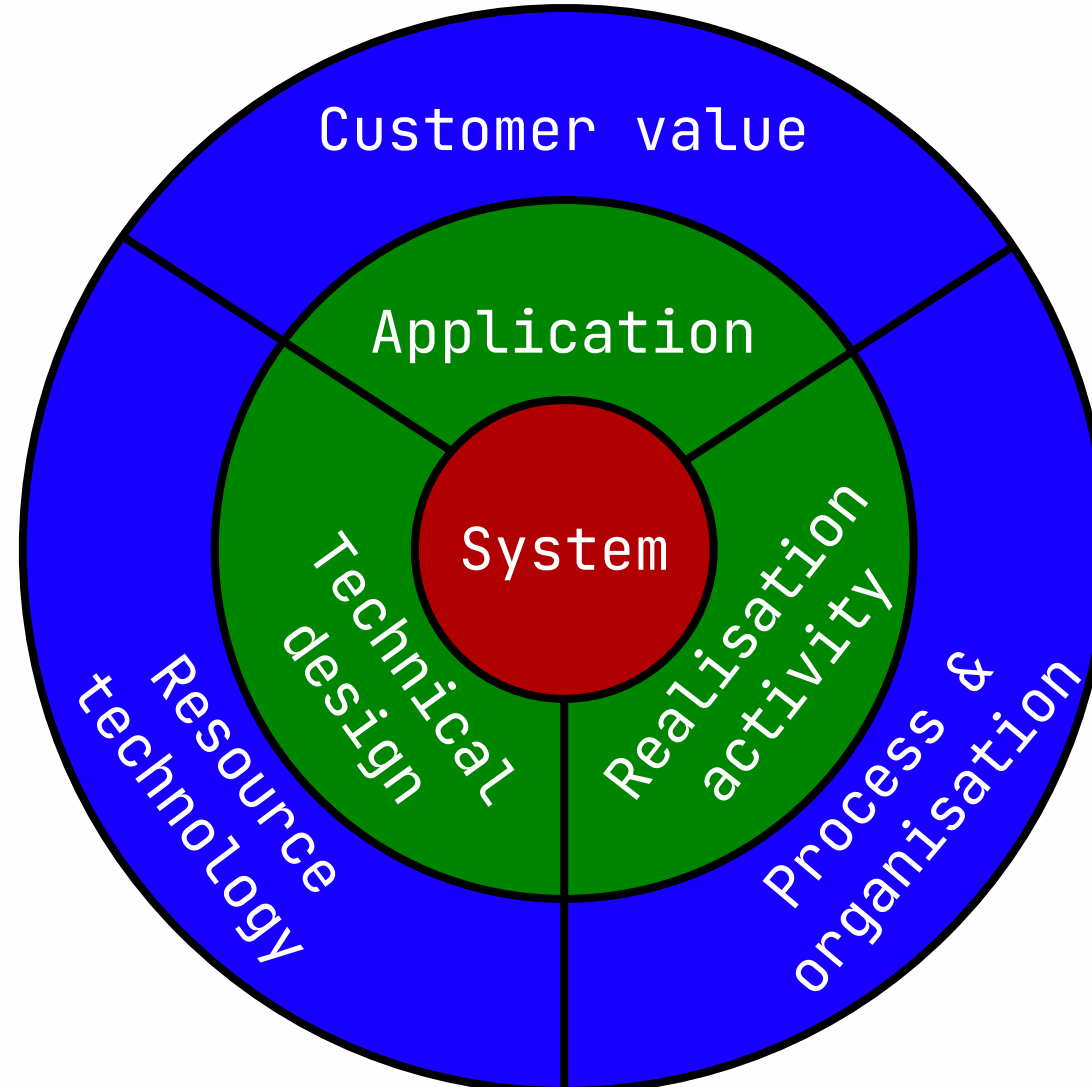


Realisation

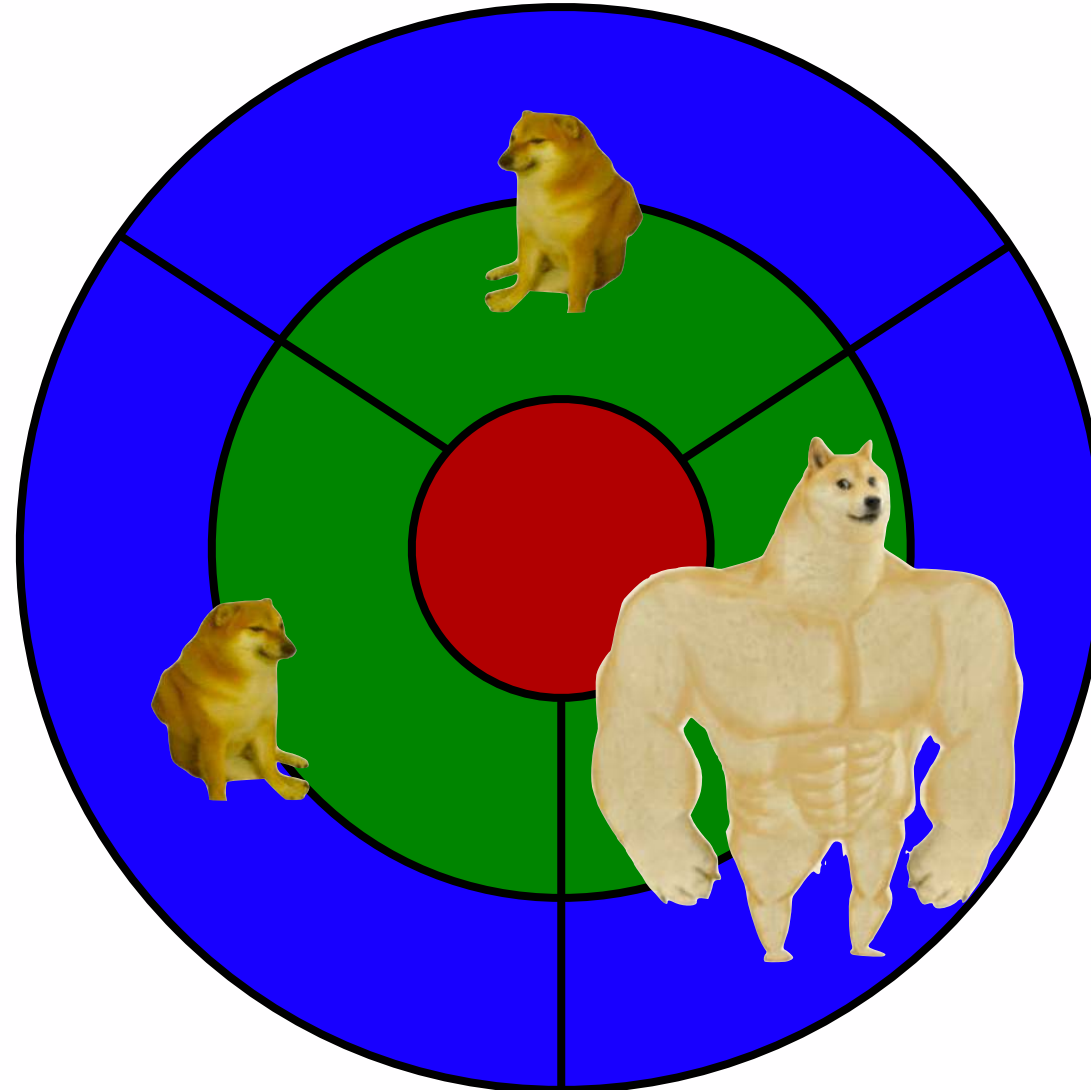
Design of Software Architectures

Dr. Vadim Zaytsev aka @grammarware, 22 September 2023

Recall this:



Recall this:



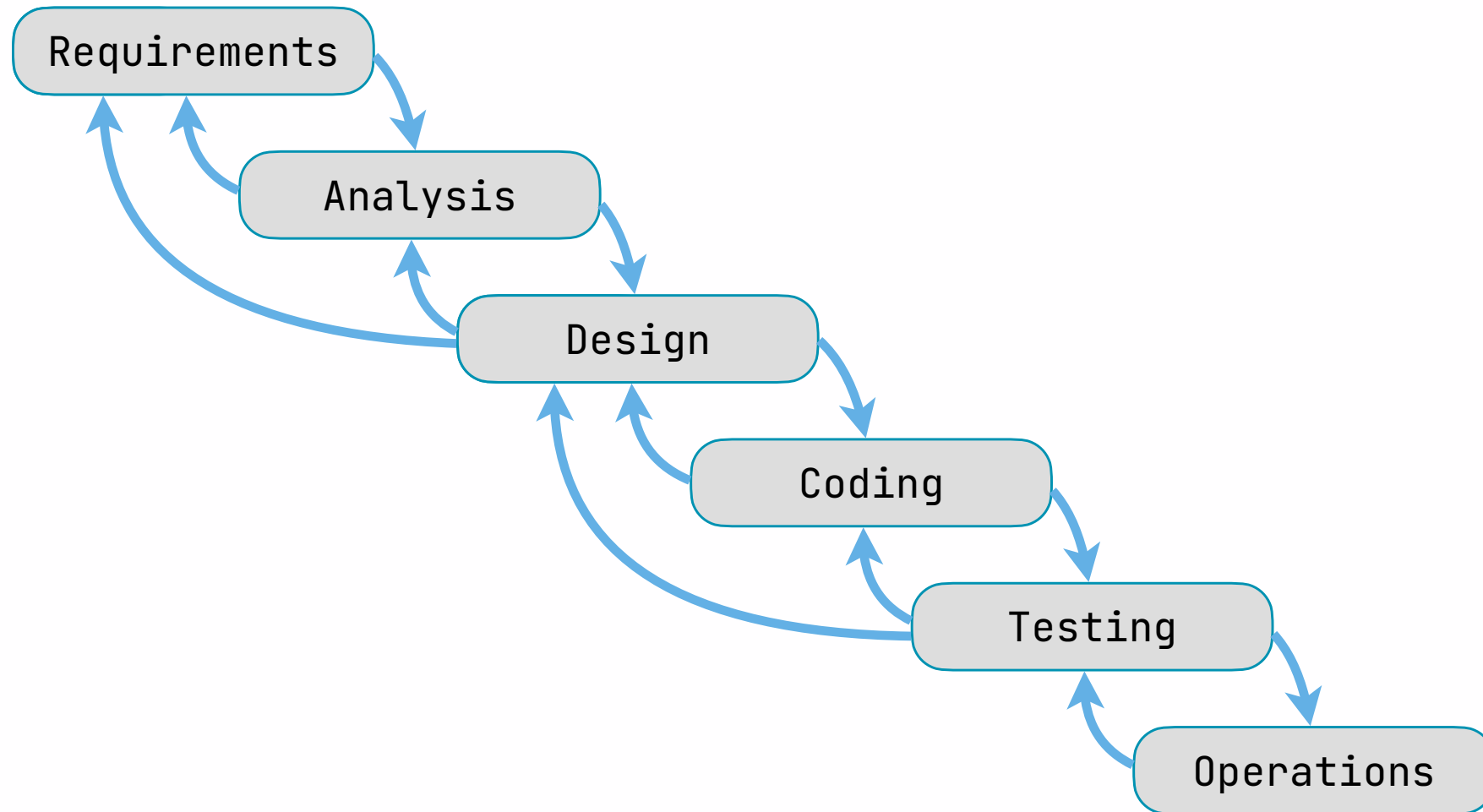
by Robert Deckers, based on ISO 42010 & DYA

Recall realisation aspects

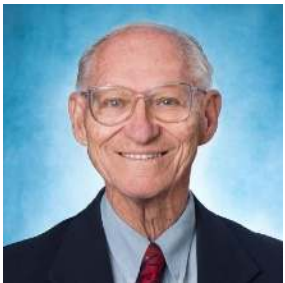
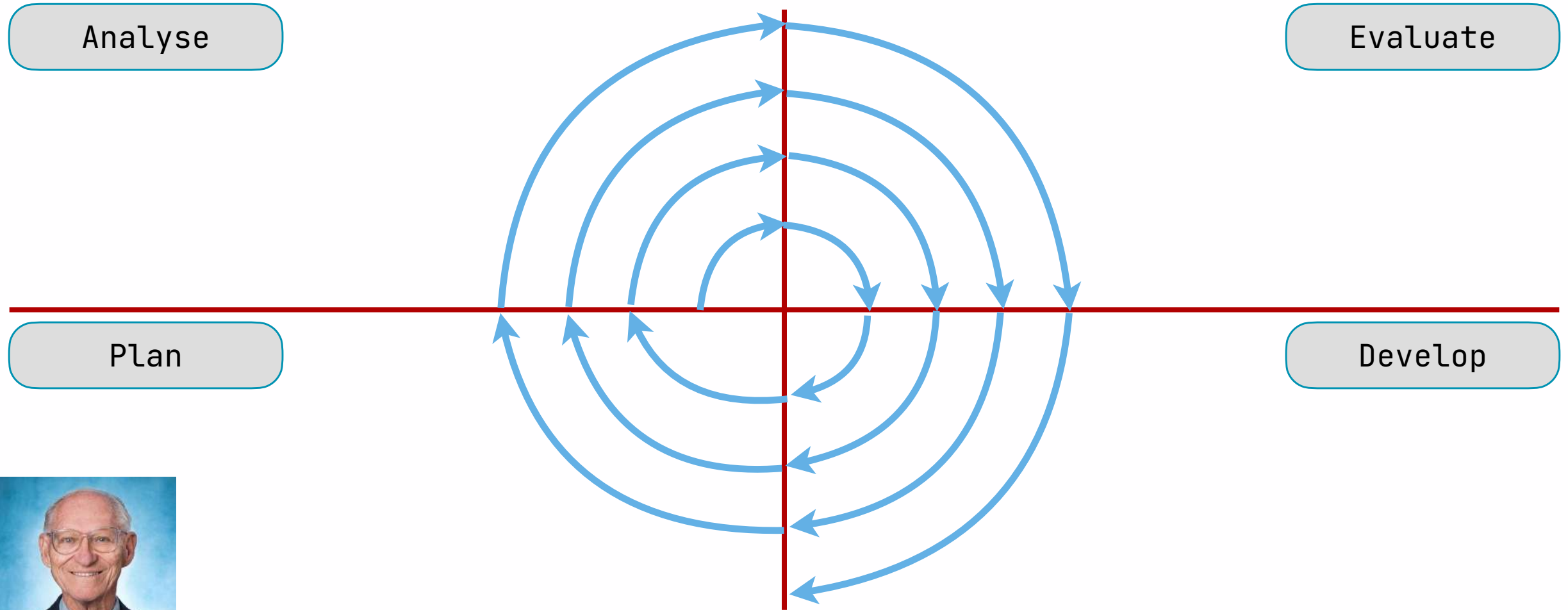
- Waterfall \iff Iterative
- Manual \iff Automated
- Buy \iff Reuse
- Adapt \iff Remake
- Feature first \iff Market first
- Develop \iff Outsource
- Assign \iff Hire
- Test for release \iff Test for integration

Think of activities

Think of activities



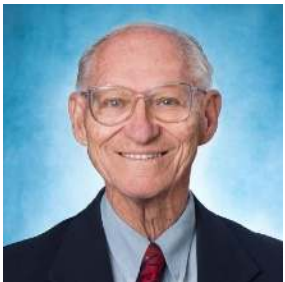
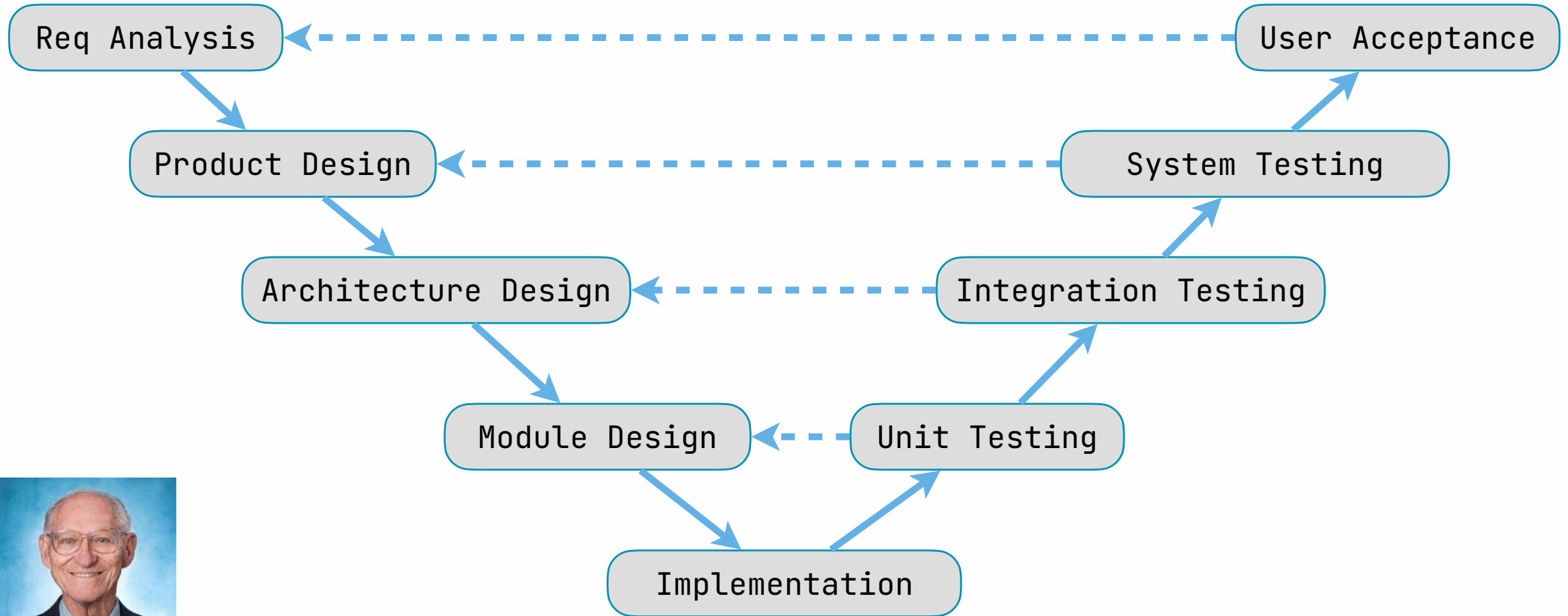
Think of activities



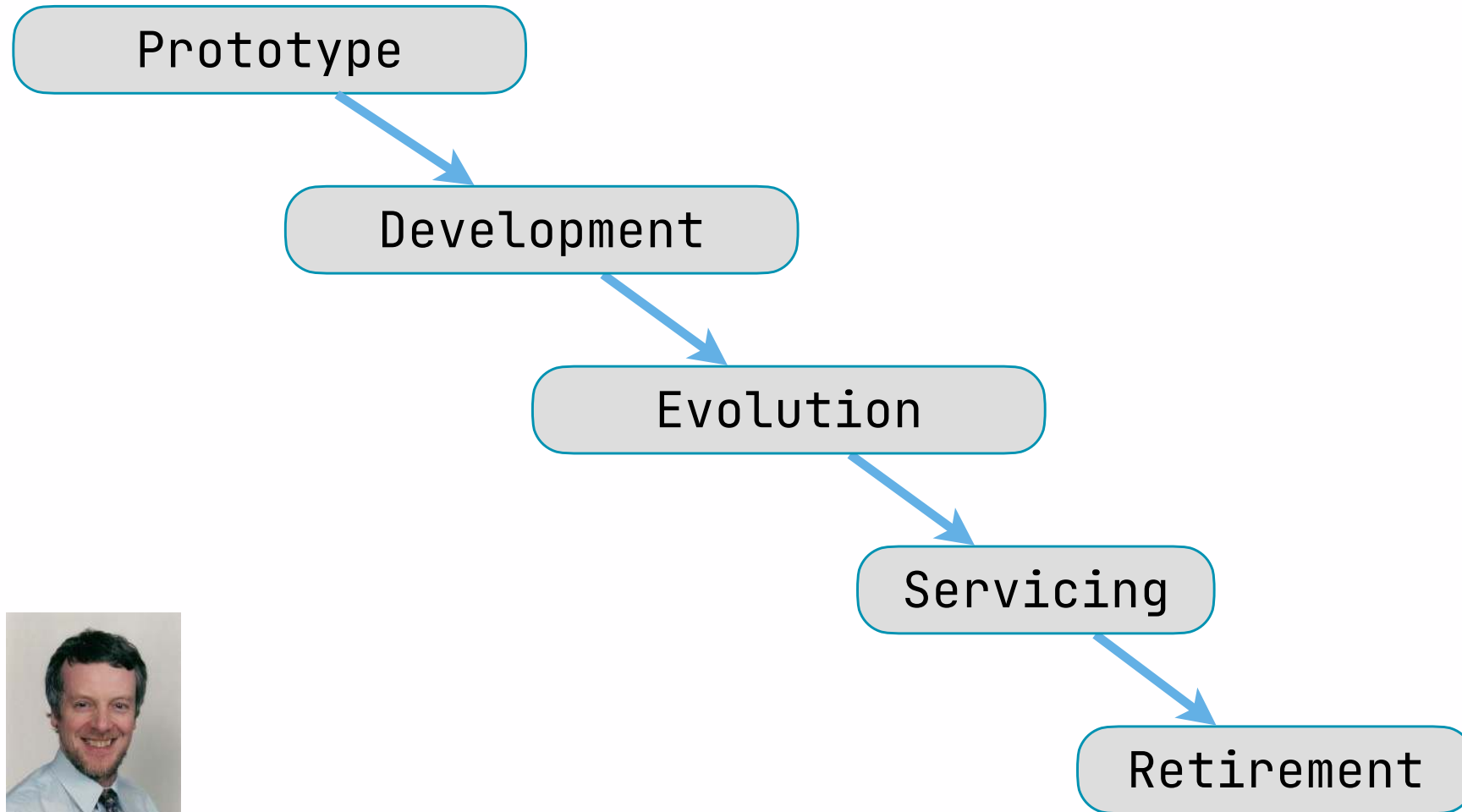
B.Boehm, A [REDACTED] model of software development and enhancement, ACM SIGSOFT SE Notes, 1986.



Think of activities



Think of activities



Think of artefacts

- Source code
- Executables, binaries
- Logs, traces
- Tests, models, stories
- Design, diagrams
- Documentation
- Packaging, installation
- Demonstrators, MVPs, PoCs, pilots
- Progress, backlog
- Feedback on input

how to organise this?

what process to follow?

Think of constraints

- Stakeholder constraints
- User requirements
- Operations / infrastructure
- Enterprise architecture
- System architecture
- Company guidelines
- Regulations
- Laws

Think of development standards

- Coding standards
- Design patterns
- Implementation patterns
- Templates
- Components
- Platforms
- Libraries
- Tools



Think of systems and people

- Systems
 - Often operate on **existing** systems
 - **Interacting** systems
- People
 - Skills and knowledge
 - Preferences and beliefs

Map architecture onto the process

- `Activities, artefacts, constraints, standards, systems, people`
- What process comes after your architecture document:
 - main `steps`
 - involved `roles` (teams, stakeholders)
- How is the architecture used by the development process?
 - Which of your `views` is used by whom?
 - How is the architecture applied `properly`?
 - How can you `verify` if it is?



