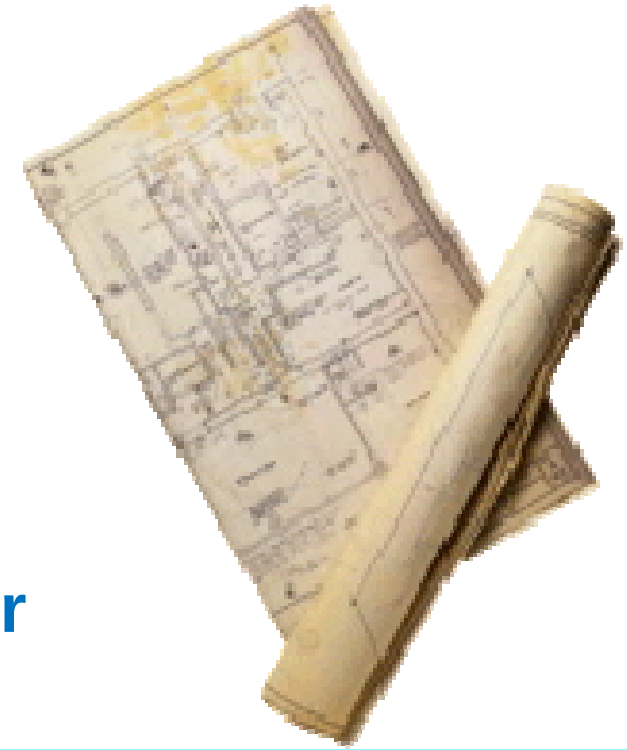
Becoming an Architect

Overview

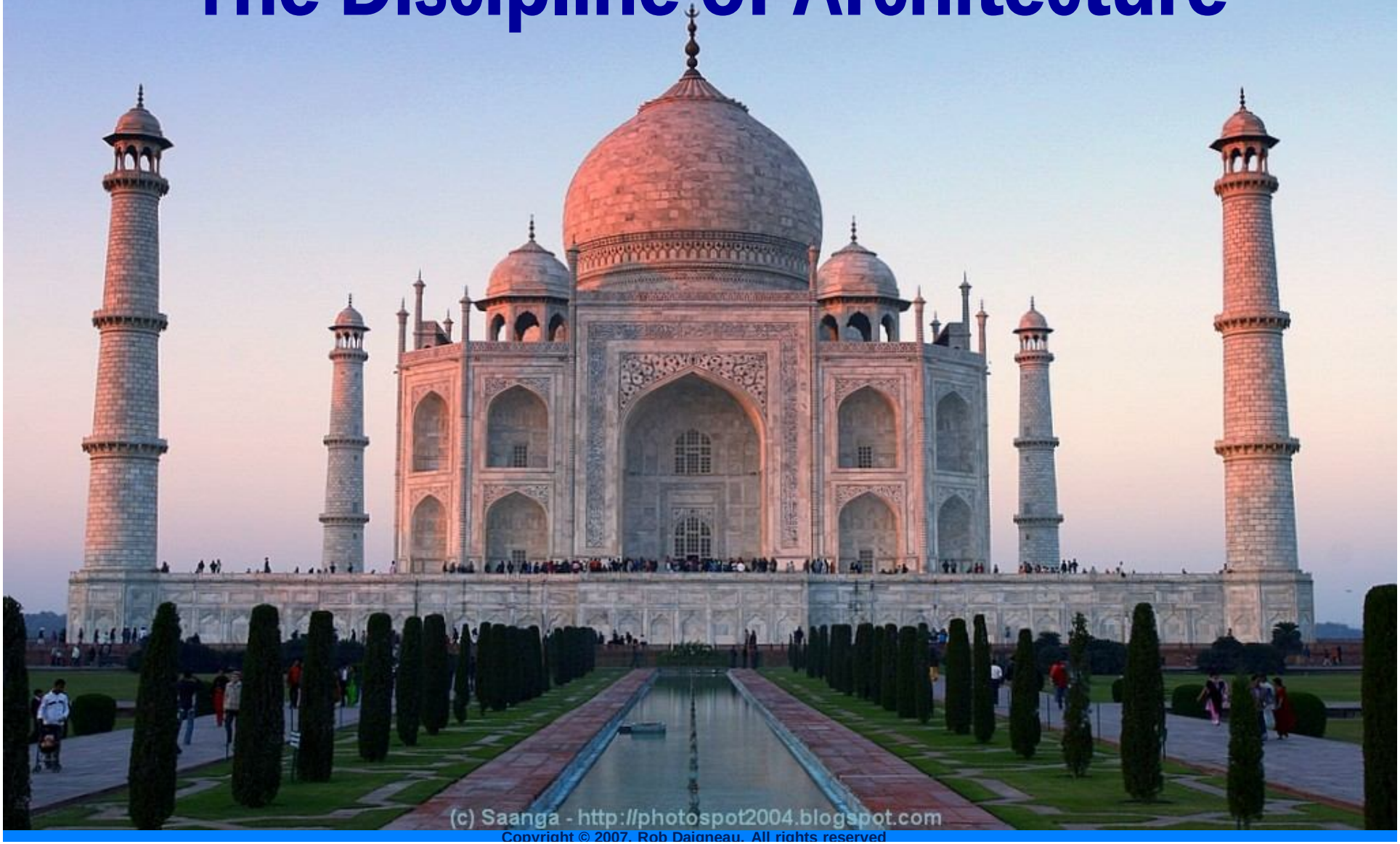
- The Discipline of Architecture
- Architectural Decisions and their Consequences
- The Realm of Architectural Planning
- The Architect's Struggle
- The Skills Required to do the Job
- The Passage from Developer to Architect



About Me

- Principal of ArcSage
- **Addison Wesley** author
  - Currently working on ***SOA Design Patterns*** book
- Host of **www.DesignPatternsFor.Net**
- Chief Architect synXis Sabre Holdings
- Director of Architecture monster

The Discipline of Architecture



What Does It Mean To Be An Architect?

No agreement

Let's start here ...

1. A person who designs **buildings** *and advises in* **their construction**.

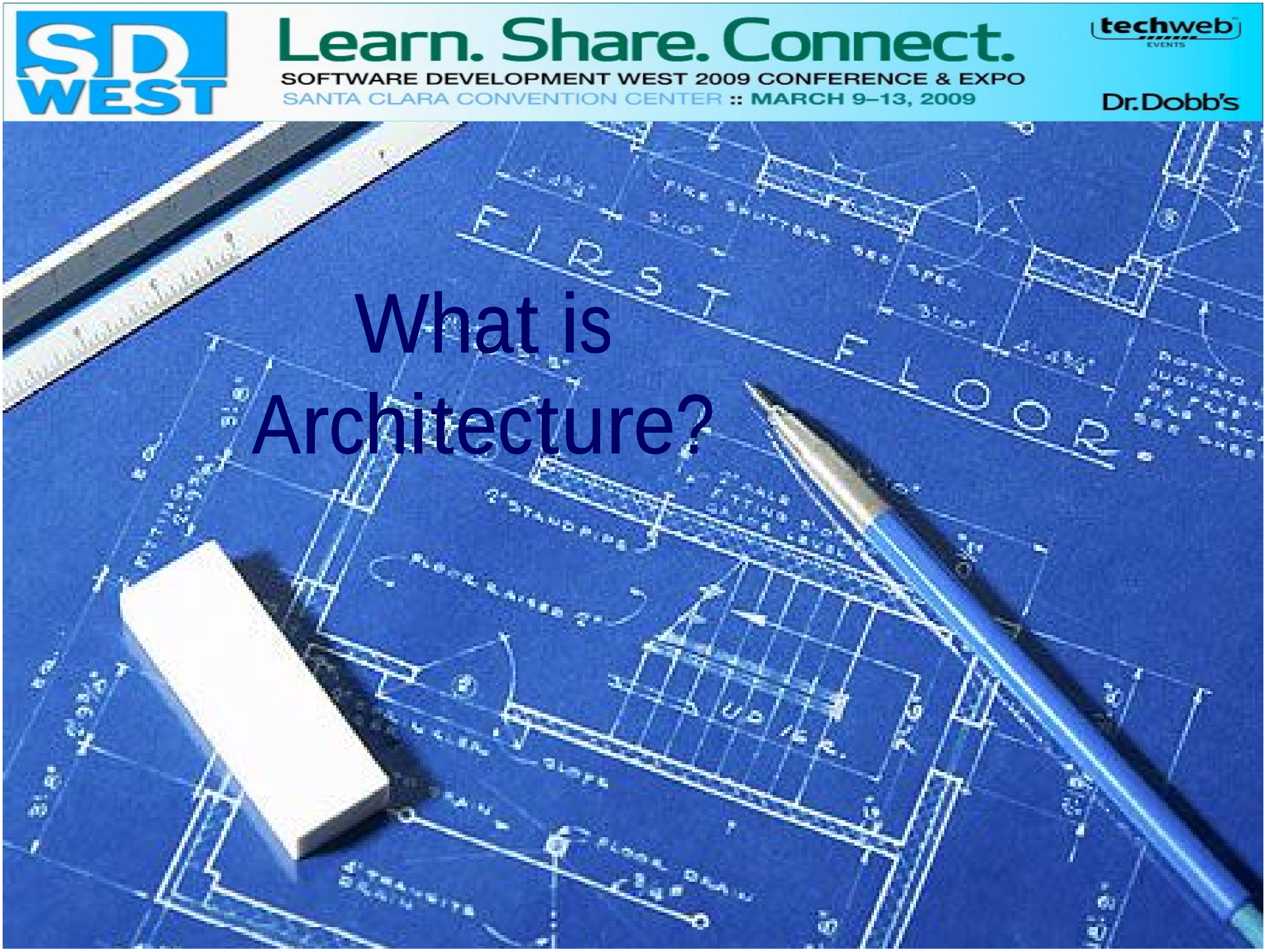
2. A person who designs and guides a plan or undertaking.

From Webster's Dictionary

Let's replace a few words …

1. A person who designs **software systems** *and advises in* **the design of such systems**.

2. A person who designs and ***guides a plan*** *or undertaking related to* **software systems design**.


SD WEST
Learn. Share. Connect.
SOFTWARE DEVELOPMENT WEST 2009 CONFERENCE & EXPO
SANTA CLARA CONVENTION CENTER :: MARCH 9–13, 2009
techweb EVENTS
Dr.Dobb's
What is Architecture?

IEEE 1471-2000

Fundamental **organization** *of a system*

embodied in its **components**,

their **relationships** *to each other and the environment*,

and the **principles** *governing its design and evolution*

Booch, Kruchten, Reitman, Bittner, Shaw

the set of **significant decisions** *about the organization of a software system*

Selection of the **structural elements** *and their* **interfaces** *by which a system is composed*

Behavior *as specified in collaborations among those elements*

Architecture is generally concerned with design at higher levels of abstraction …

- Emphasis is on general organization, relationships
- **Less focus on "the How's"**

… Architecture might also look at "lower level concerns"

- These types of architects may look at "the How's"
  - Detailed design
- The altitude at which architects fly can be contentious

The Many Species of Architects

Solution Architect

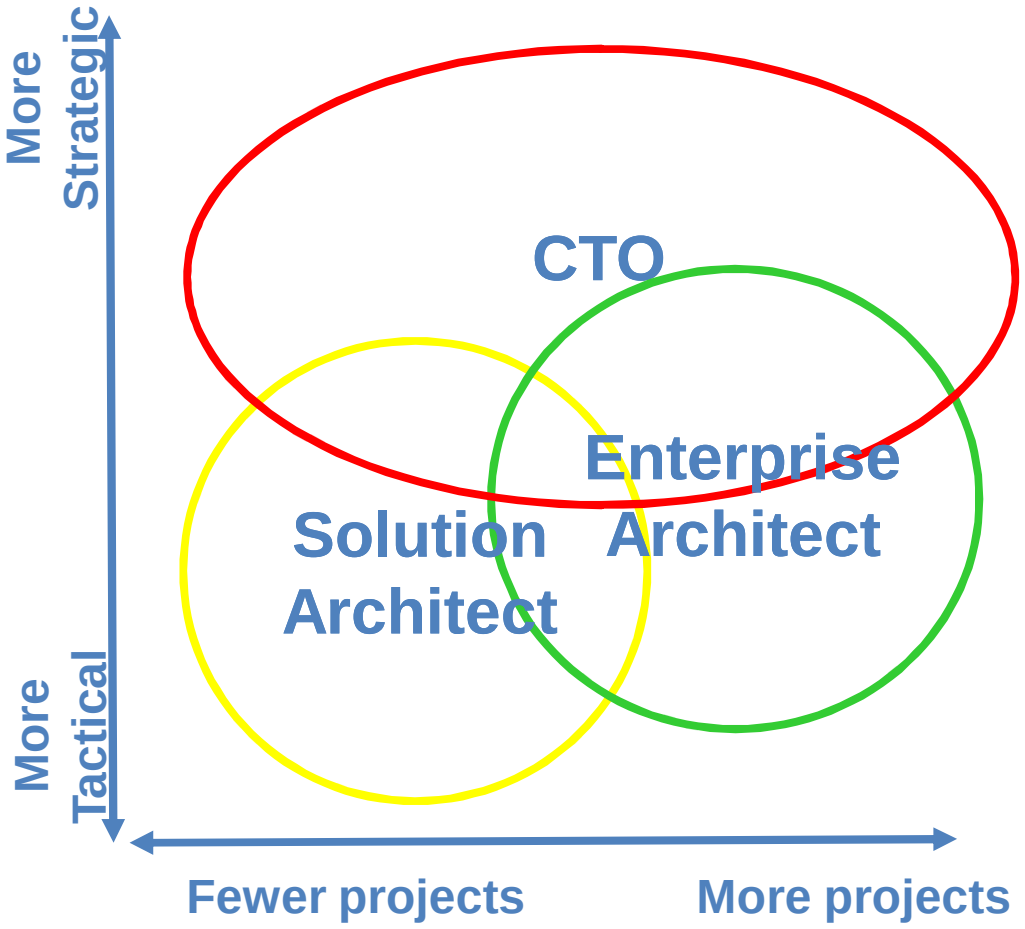
- Typically leads design of one or a few products
- Tends to be more "hands on"

Enterprise Architect

- Aligns use of technologies with strategic business initiatives
- Drives for consistency in product design, methodologies, tools, etc., across the enterprise
- Provides oversight

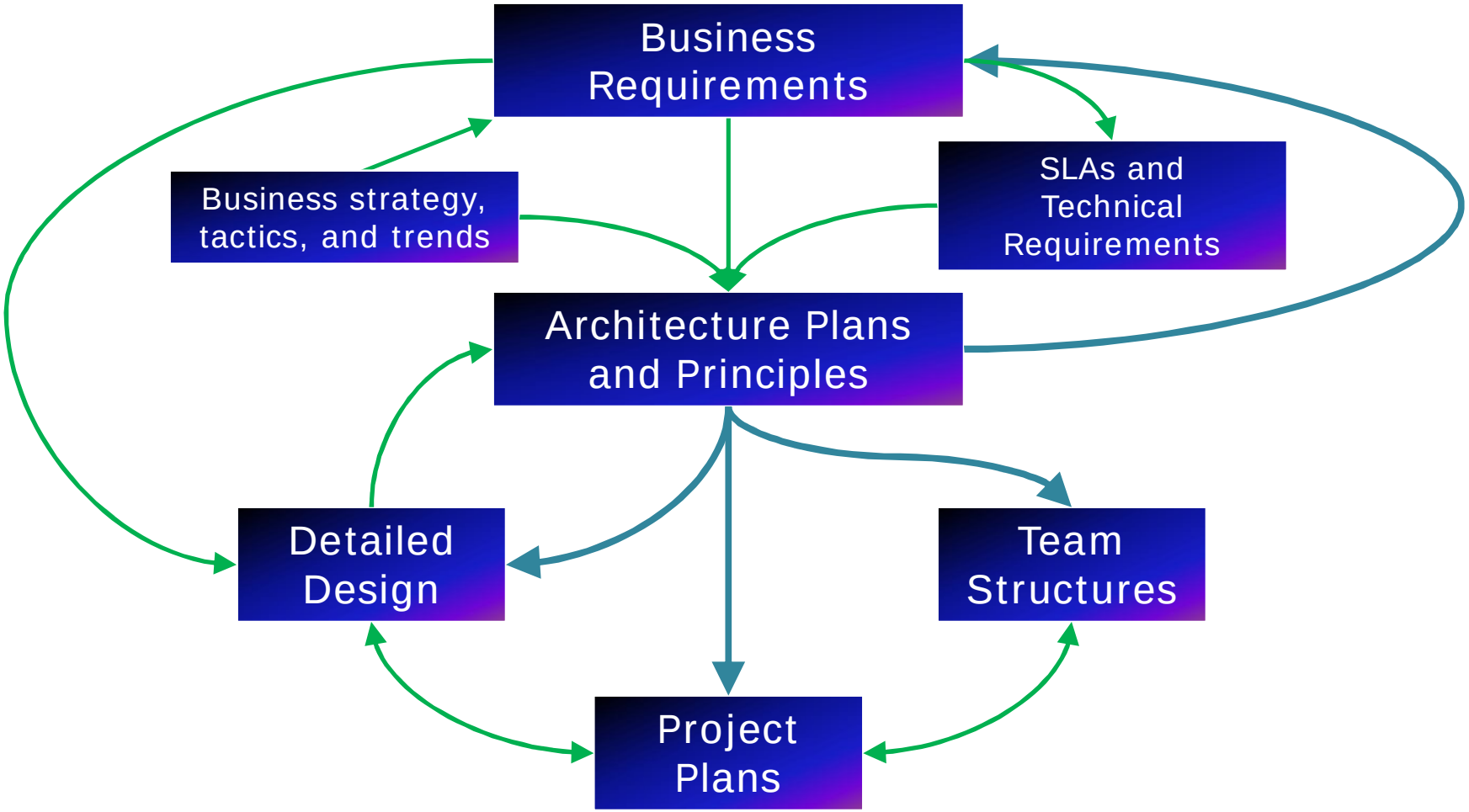
CTO

- Highest level of oversight
- Manages budgets, SLAs, and contracts

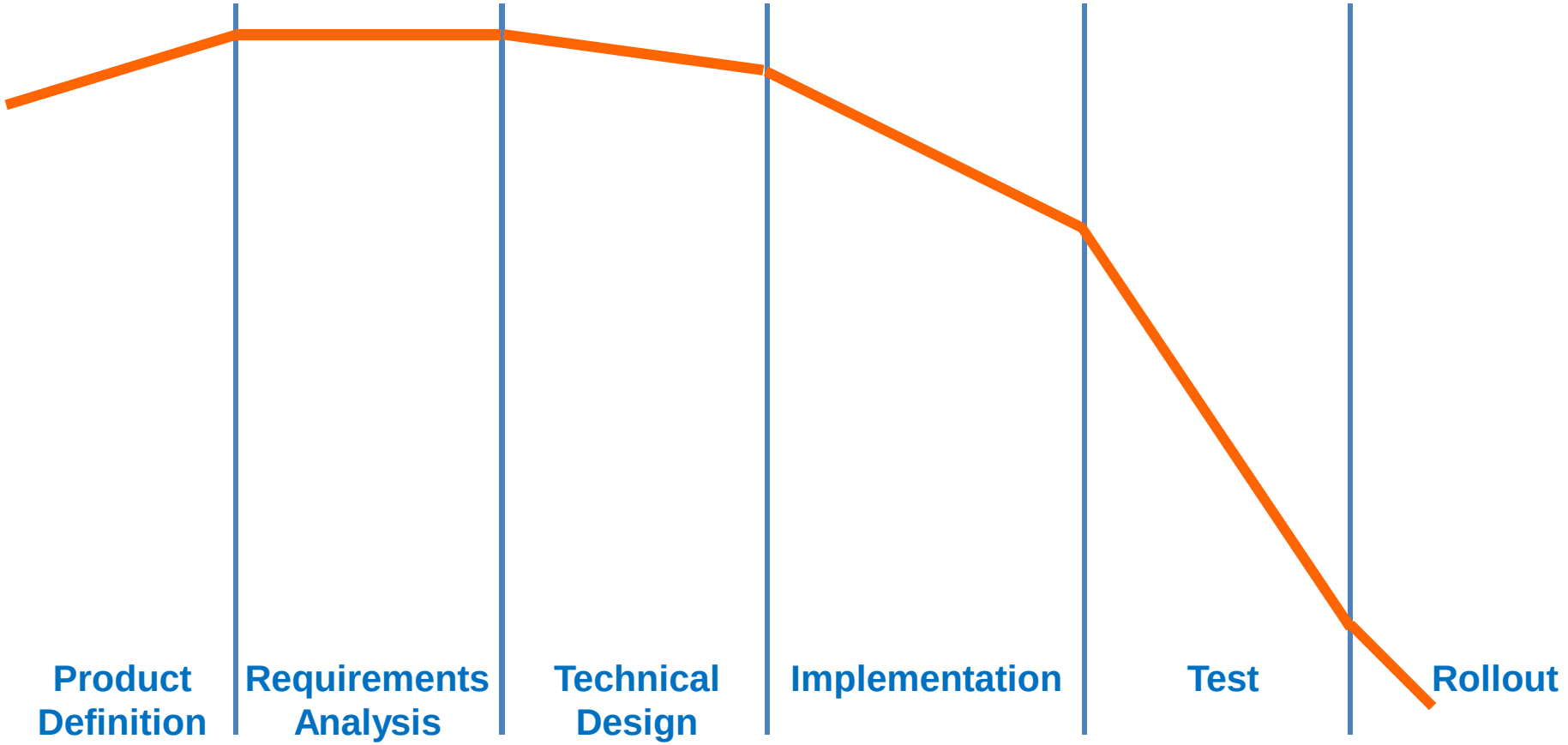


- **Other types of architects: e.g. Infrastructure, Data, Information, etc.**

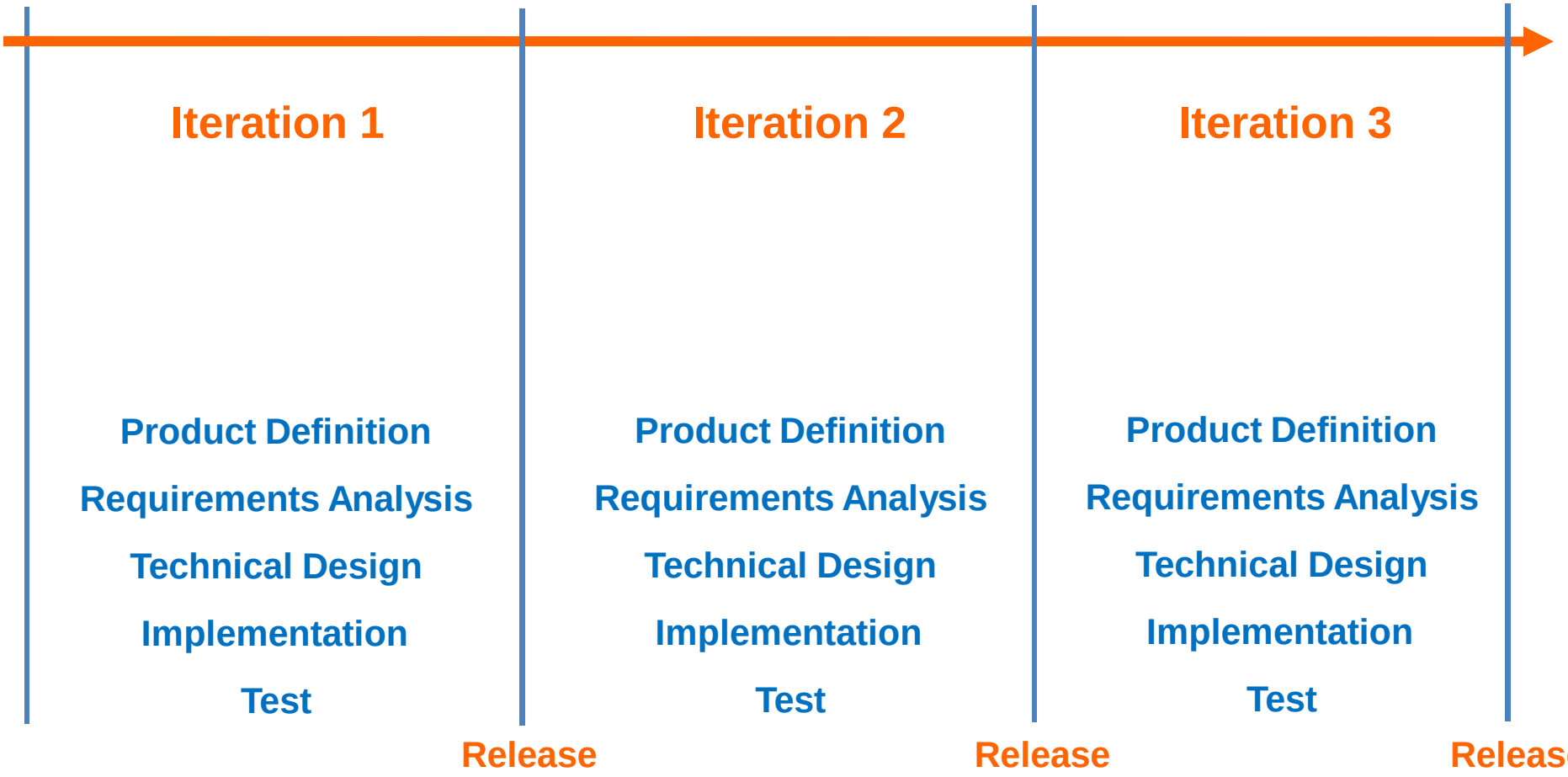
The Influence of Architecture



When Do Architects Typically Get Involved? The Traditional View



When Do Architects Typically Get Involved? The Agile View



Architectural Decisions and their Consequences

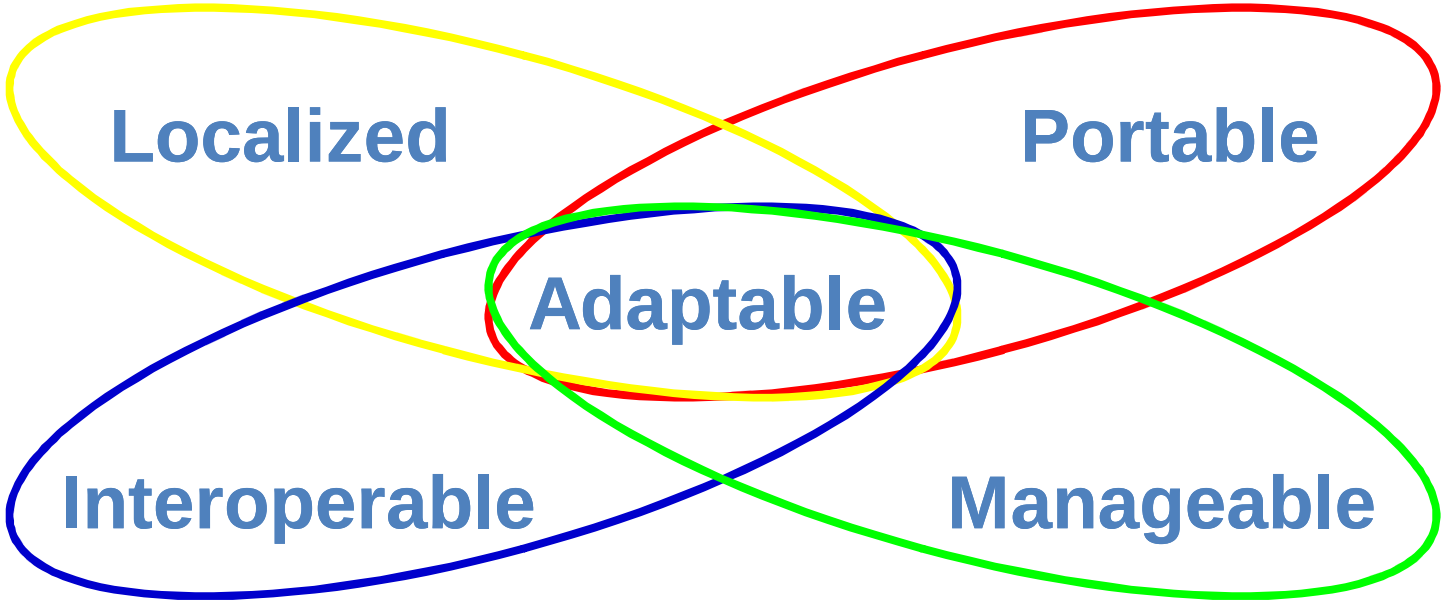
Quality of Service Attributes

A type of technical requirement …

- Adaptable
  - Maintainable, Extensible
- Efficient
- Interoperable
- Manageable
- Available
- Responsive
  - i.e. Performance (e.g. Throughput, Response Time), Scalable
- Robust
- Secure
  - Authentication, Authorization, Privacy

Compatible Goals

Many goals do not oppose each other, e.g. …



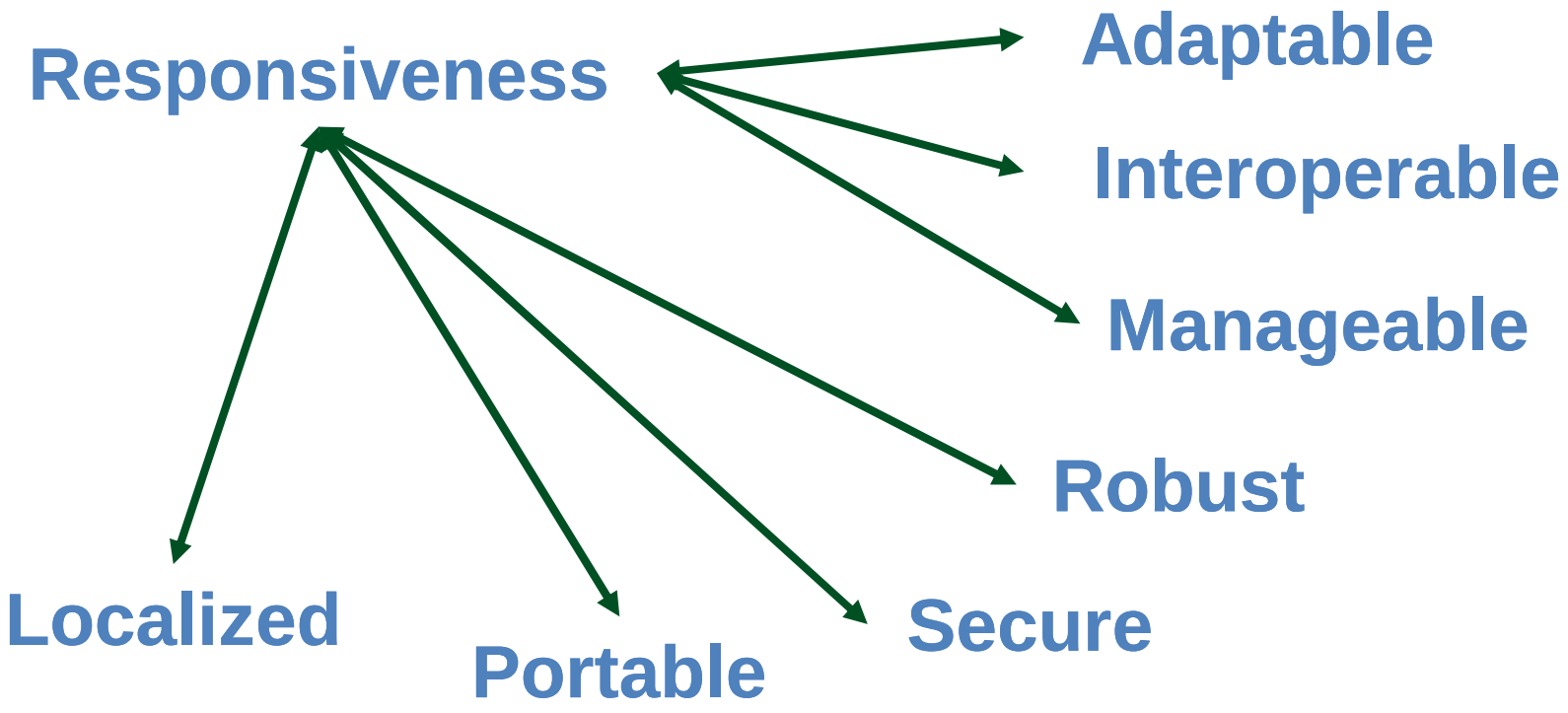
You Can't Have it All

- All architectural decisions have positive and negative consequences
- A movement to optimize one thing may cause a trade-off on something else

Evaluating Trade-offs Part 1

Evaluating Trade-offs Part 2

Some goals naturally compete against each other



The Realm of Architectural Planning

Guiding Principles and Standards

Key concepts to guide …

- Selection of technologies
- Decisions made for
  - Detailed physical design
  - Deployment/Execution/Operations architecture

Influenced by

- Business and technical requirements

Examples

- QOS Attributes to optimize for
- Buy/Borrow/License vs. Build Philosophy
- When/where to use open source
- Coding guidelines and standards
- Use of Agile techniques

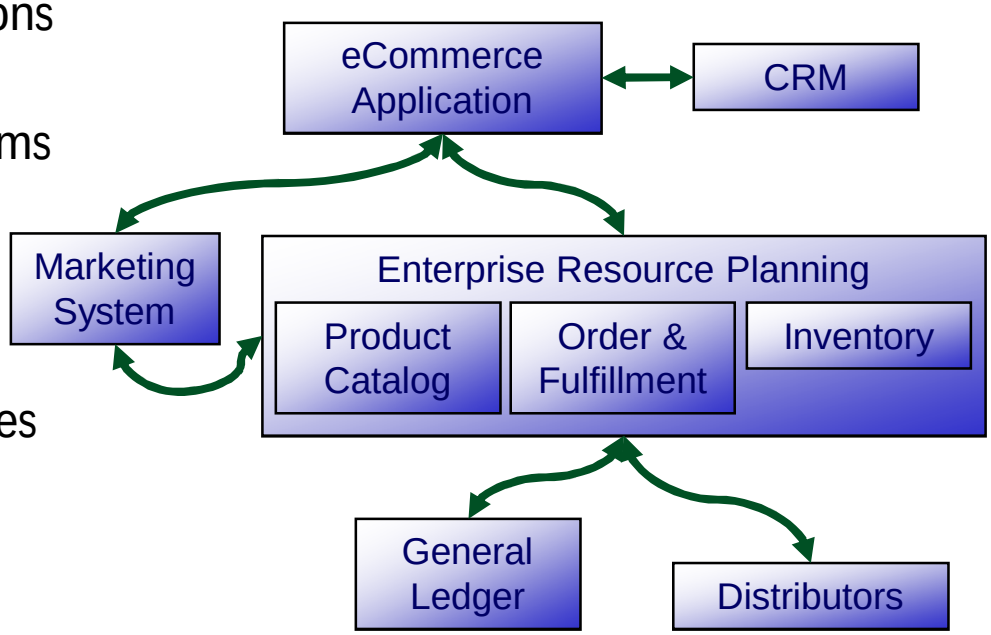
Conceptual Blueprints

High level description of

- Major subsystems and Integrations
- Shows how business capabilities mapped to subsystems

A strategic view

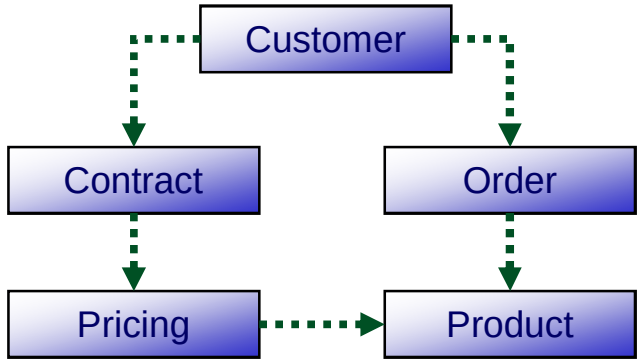
- Shows evolution of architecture over time in several diagrams and narratives
- May look out several years
- **Does this conflict with Agile?**



Functions as a communication device

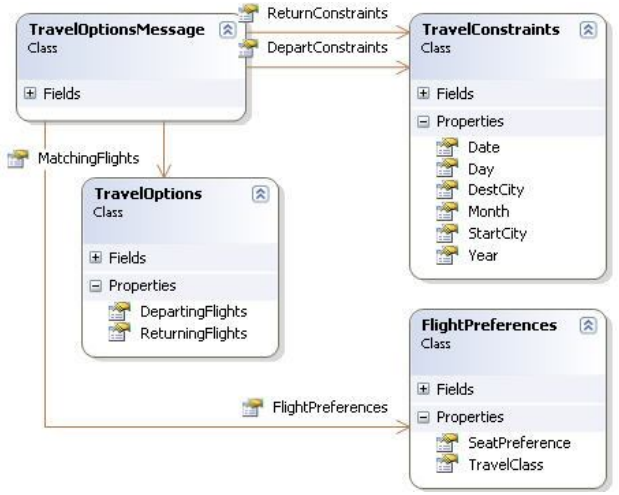
Logical Models

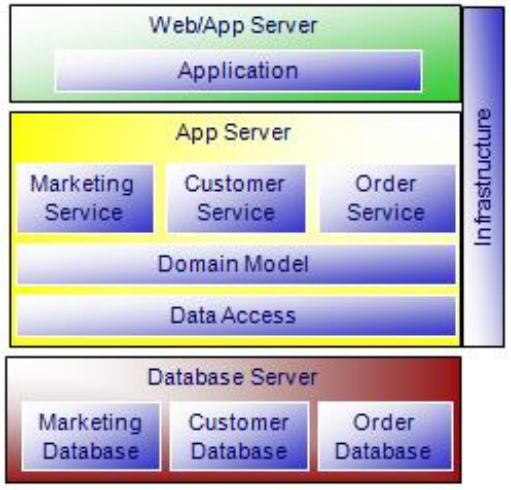
- Defines High Level …
  - Structures, relationships, responsibilities, behaviors
  - **Platform-independent**
- Why?
  - An analysis tool
    - Helps business and technology stakeholders get better understanding of problem domain
    - Identify flaws, issues, and challenges early
    - Explore alternatives
  - Preparation for physical design
  - Supports use-cases or user stories
- How ?
  - Domain-Driven Design, CRCs, Workflows



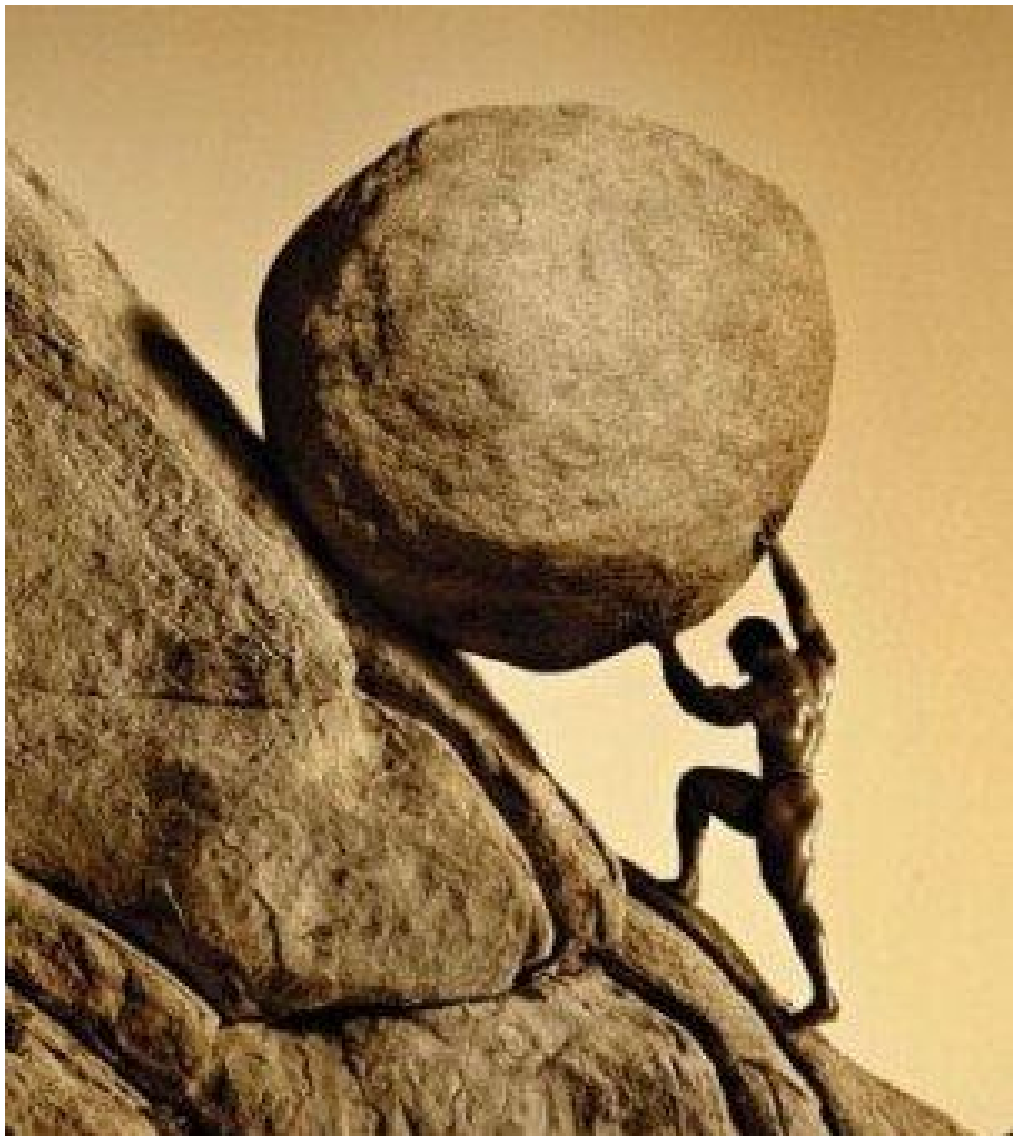
Physical Design

- What many (*developers*) think of when they consider architecture
- Platform-specific detailed design
  - Services, major interfaces, classes
  - Design patterns
  - Integrations *(i.e. messages, events, transforms, etc.)*
  - Deployment/Execution Architecture
    - Shows how components are deployed
    - Server configurations, network topology, and capacity plans
- Agile cautions against "Big Design Up Front" (BDUF)





The Architect's Struggle



"The Ideal" will always be out of our grasp

- There is much that conspires against the Architect
- Technical elegance, consistency, etc., are oftentimes fleeting

Where the Architect metaphor breaks down

Building architecture is static,
Software architecture isn't

Spaghetti-Code is Pervasive

- Tactical initiatives usually take precedence over "Architectural Wisdom"
- "Technical Debt" rarely gets re-paid
  - Unless …
    - Something really bad happens
      - e.g. Security breach, failure to meet SLA
    - A "brick wall" is looming
    - The velocity of delivery slows to a snail's pace

    … the list goes on

Decay is Inevitable

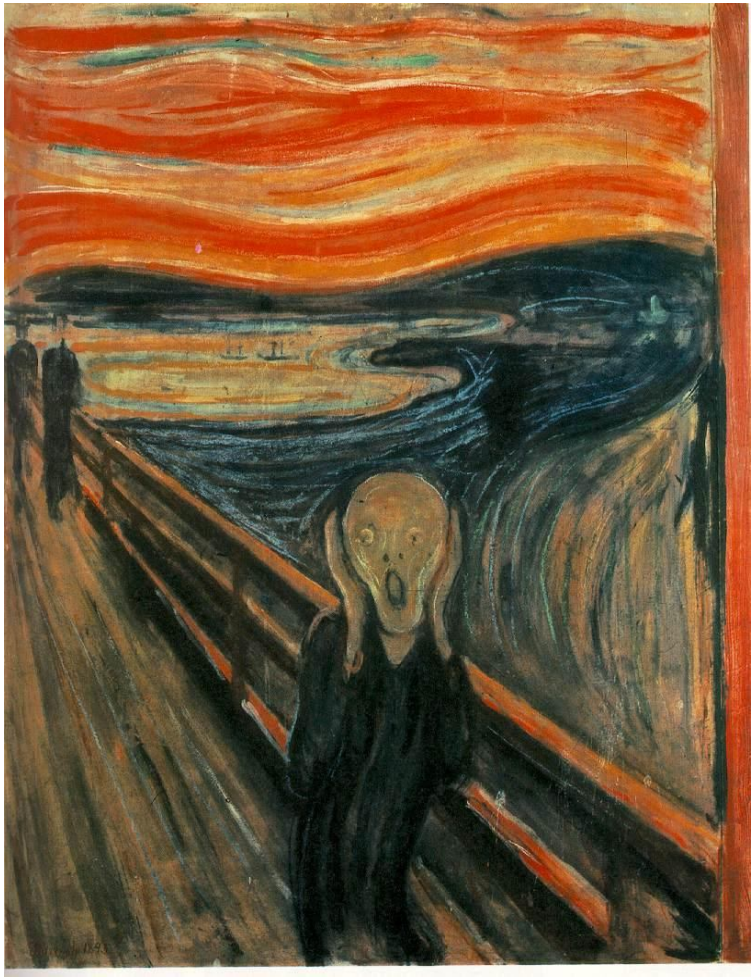
unless you do some housekeeping
…. i.e. **Continuous Refactoring**

Why Do These Things Happen?

- Time, Amount of Work/Labor
  - Which would you choose?
    - An inelegant solution that goes to market quickly?
    - An elegant solution that takes longer to release?
- Money, Expense
  - May be difficult to assign a Business Value or ROI to an architectural change
    - Believable empirical data may not be available
    - Are you placing bets? Going on intuition?
  - Pay-off might not be immediate
- Experience, Skill

The Human Factor

- People are
  - Unpredictable, irrational at times
- Different perspectives
  - No one will ever code or design something the way you would have
  - Can you really convince management regarding things they may not understand?
- Politics and personal agendas are unavoidable
  - They exist in small and large organizations alike



Herding Cats

Team challenges increase as

- Size of team grows
- Teams become more distributed
- Cultural diversity increases

The Skills Required to do the Job

The "Easy" Skills

- Abstraction
  - Business and technical concepts
- OO Design Concepts
- Analysis and Modeling Techniques
  - e.g. UML, CRC, ERD, etc.
- Relational Databases
  - Normalization, Transaction and Concurrency Management, Data Integrity
- Design Patterns
- SDLC Methodologies
  - Prescriptive (e.g. Waterfall, RUP) vs. Adaptive approaches (e.g. Agile)
- Project Management Basics
  - Task planning, schedule estimation, monitoring, reporting, risk management, budget management, etc.
- Knowledge of your Business Domain

The Hard Skills are the Soft Skills

- Mentoring, Consulting, Coaching, Delegation
  - Wise architects let people choose their own paths
  - Architects shouldn't drive everything
  - Wisdom acquired from own mistakes

The Hard Skills are the Soft Skills

- Persuasion, Influence
  - Architects may not have authority
- Politician, Negotiator,
  - Must be pragmatic, able to compromise
  - Learn who the players are
    - Who defines what's important?
    - Who leads, who follows?
    - How do you make them successful ?

Develop Patience, Let Go

- Seek to understand "their" Fears, Uncertainty, and Doubts (i.e. FUD)
  - Empathize, develop trust
- If you're trying to introduce change …
  - Recognize that progress might only be made in inches rather than miles
  - Every little step in the right direction is a good thing.
- Learn how to accept chaos, disarray, and spaghetti
  - Consider what "Good Enough" software means to you

Move Wisely

- How, where, and when do I insert myself for maximum effect?
- Align yourself with the company/business agenda
  - Usually about gaining market share, meeting contractual obligations, reducing costs, etc.
- Know when to adapt or even abandon an idea
  - Politics and personal agendas can sideline even the best rationalized ideas
  - Recognize that all systems and applications have a useful lifetime

The Passage from Developer to Architect

Architects and Developers

Effective Architects are full members of the development teams

The Two Types of Architects

COMMITMENT

The chicken is involved. The pig is committed.

Should Architects Code?

It depends …

- What kind of Architect are you?
- How big is your company, your project, and your team?

Are Solution Architects actually Lead Developers?

- a.k.a. Application Architect
- Emphasis for these architects may be more on detailed design
- May not pay as much attention to other aspects of Architectural planning

Perhaps we should be asking "How much should architects code?"

Why It's Good To Keep Coding

… or staying in shape

- Abstractions will only take you so far
  - Many developers need to see code
  - Code helps to find flaws
- Helps to tear down the Ivory Tower
  - Increases credibility
  - Helps you understand developer perspective
- Allows you to cut through vendor hype

A Parable from Star Trek

- The senior-most officers couldn't delegate
  - Were always members of the landing parties

Dangers

- When you're on the front-line, who's watching to see that the ship stays on course?
- A problem of bandwidth

Architects Should Be Like Flight Directors

Mission Control had many senior engineers

The Flight Director

- Coordinated work of specialists
  - Solicited recommendations and decisions
  - Didn't do their work
- Made the final call when necessary
- Was **accountable**

What's Happening to Me?

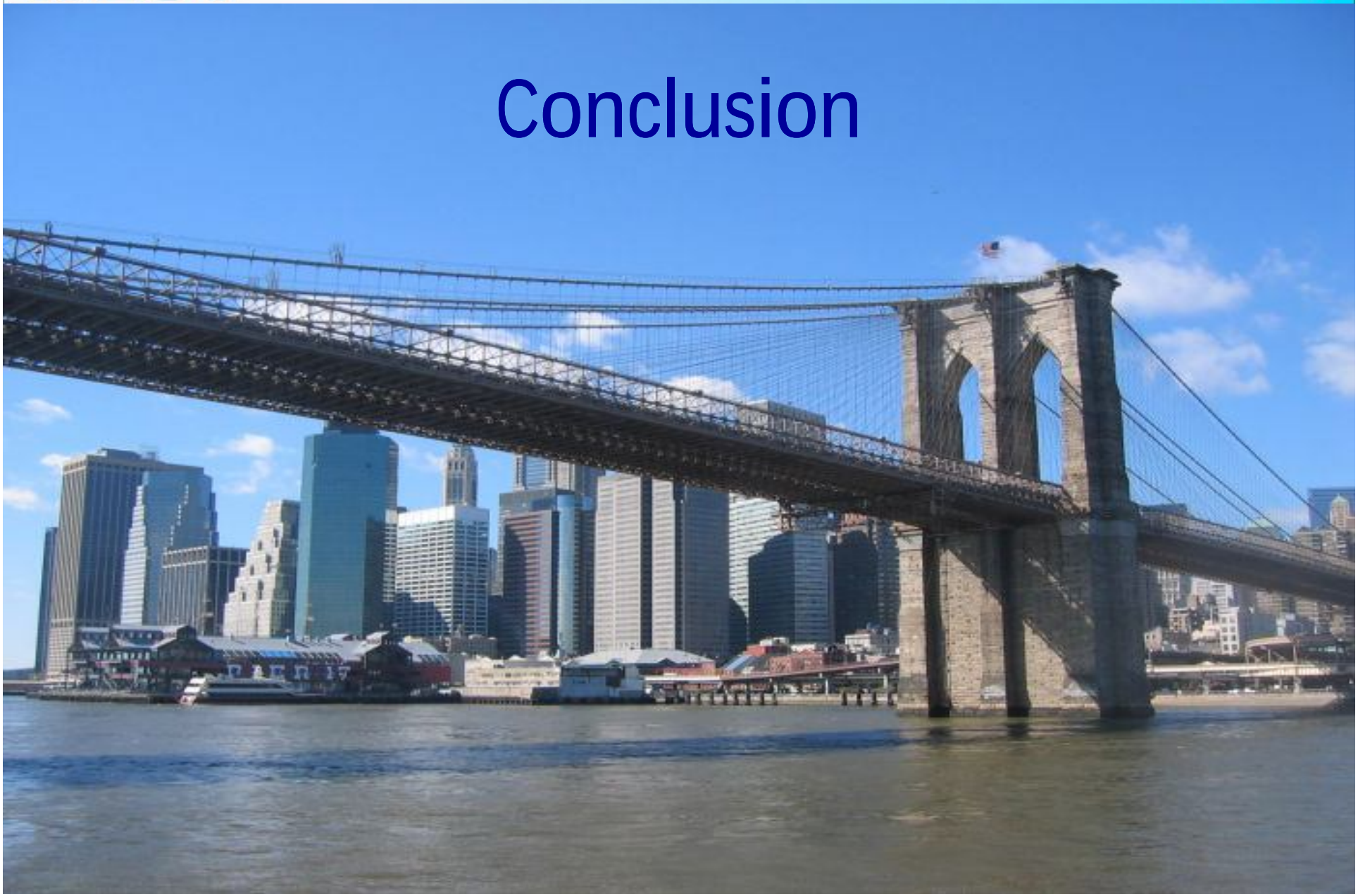
Changes …

- Keeping up with trends, changes in languages, frameworks, tools, etc. will be harder
- Your coding skills may slip
- Will probably be in more meetings and have to deal with more politics

Let Go

- Resist impulse to jump in because you think you can get it done faster

Conclusion



Recommended Resources

- www.DesignPatternsFor.Net
- The Secrets of Consulting
  - Gerald Weinberg, Dorsett House, 1985
- Rapid Development, Taming Wild Software Schedules
  - Steve McConnell, Microsoft Press, 1996
- Dynamics of Software Development
  - Jim McCarthy, Microsoft Press, 1995
- Leading Geeks
  - Paul Glen, Jossey-Bass, 2003
- The Mythical Man Month
  - Fred Brooks, Addison Wesley, 1975
- Extreme Programming Explained
  - Kent Beck, Addison Wesley, 2005
- Silos, Politics, and Turf Wars
  - Patrick Lencioni, Jossey-Bass, 2006
- Big Ball of Mud
  - Foote, Yoder, http://www.laputan.org/mud/
- Technical Debt
  - Ward Cunningham http://en.wikipedia.org/wiki/Technical_debt