Developer to Architect

The Software Architect Role in the Enterprise

Chris Simmons
www.avidsoftware.com
chris.simmons@avidsoftware.com

pluralsight
hardcore developer training

Course Overview

- **Fundamentals of being a software architect in the enterprise**
- **The Who and What of Software Architecture**
  - Who a software architect is
  - What the role entails
  - Skills, Knowledge & Duties
- **Architects role in the project life cycle**
  - Discuss each phase of the project life cycle
  - What effective architects do in each phase
- **Design**
  - Design process
  - How to effectively communicate your designs
- **Practical as possible**

The Who & What of Software Architecture

Chris Simmons
www.avidsoftware.com
chris.simmons@avidsoftware.com

pluralsight
hardcore developer training

The Who & What of Software Architecture

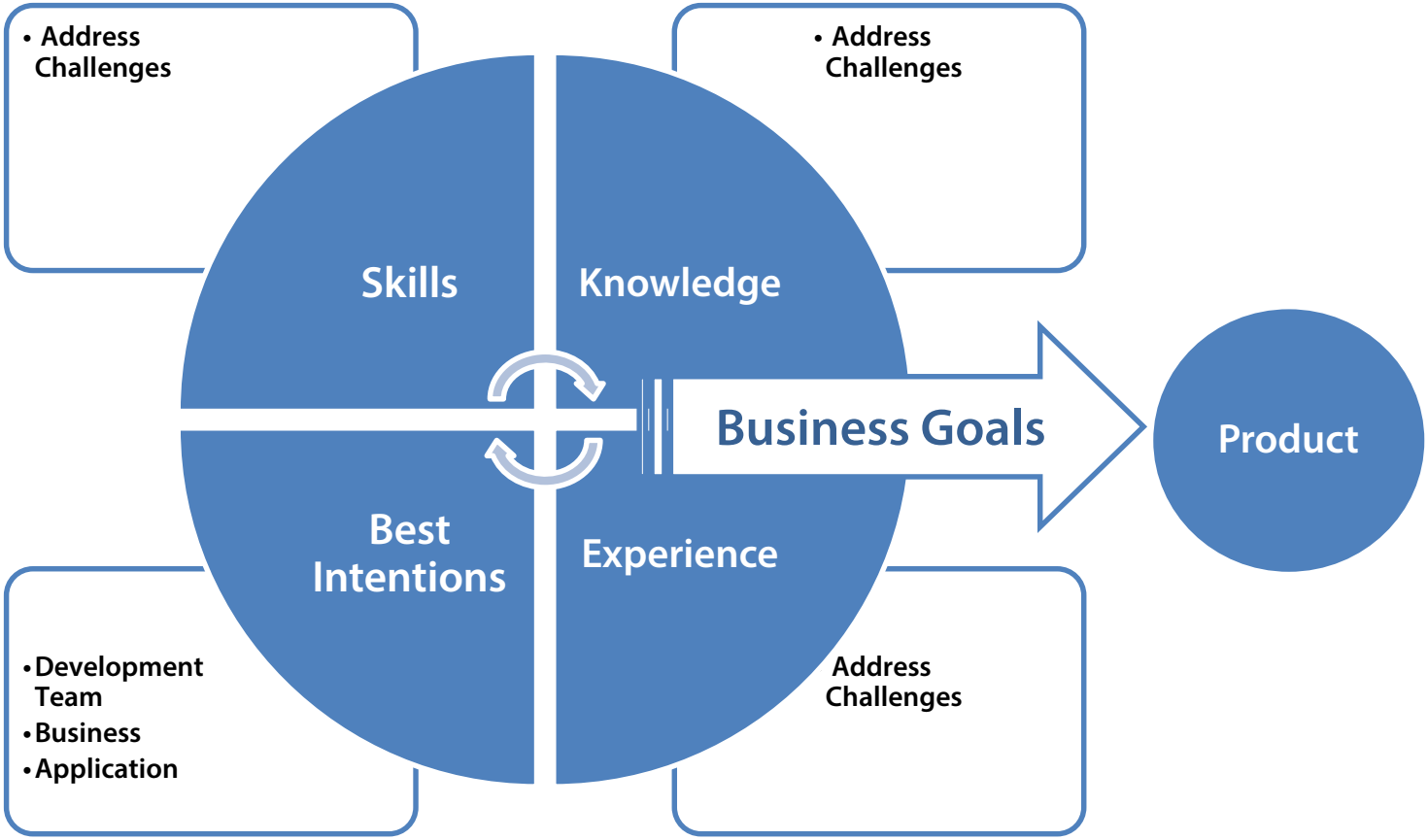
- **Software Architect role in the organization is often not clearly defined**
- **Skills**
- **Knowledge**
- **Duties**

- **Who is a software architect?**
- **What is an architect expected to be skilled at?**
- **What is an architect expected to know?**
- **What is an architect expected to do?**
  - Technical & Non-Technical
- **Why does an organization need software architects?**

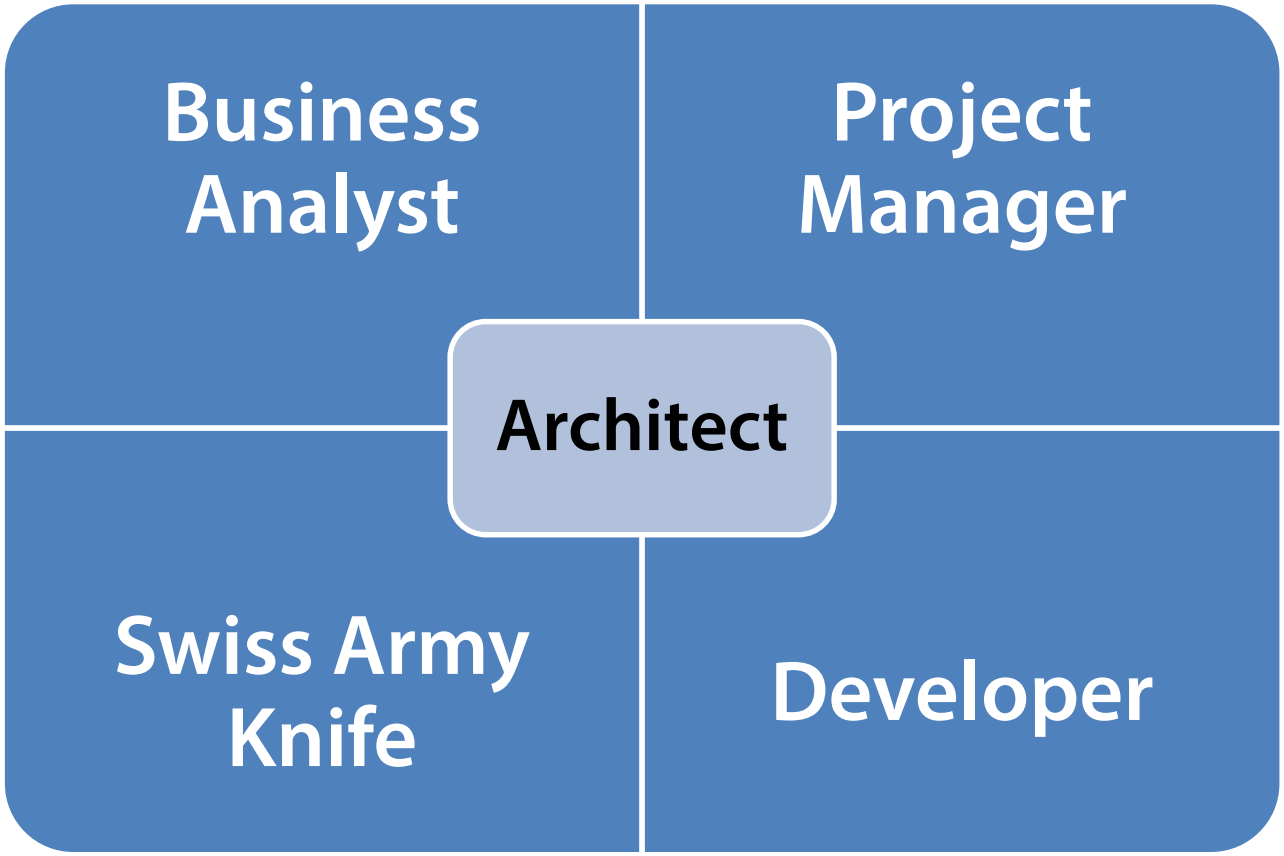
Who is a Software Architect?

- Why are your applications in the state they are in?
- Silo'd design
- Little or no reuse
- Poor design
- Why?
- Same discussion over and over
- No time
- No resources
- Need to redesign…
- Redesign is difficult
- Talk but no action
- Design but no implementation
- Business doesn't see the benefit

Who is a Software Architect?



Who is a Software Architect?

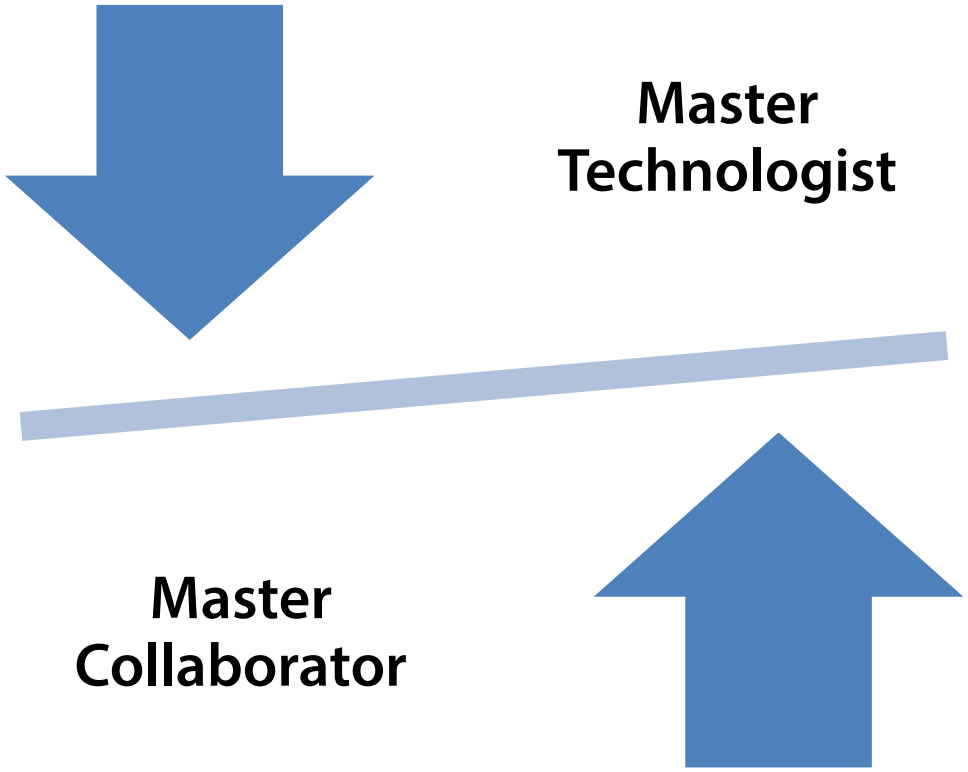


Who is a Software Architect?

“a computer programmer who makes high-level design choices and dictates technical standards, including software coding standards, tools, and platforms.”

-- wikipedia --

Who is a Software Architect?



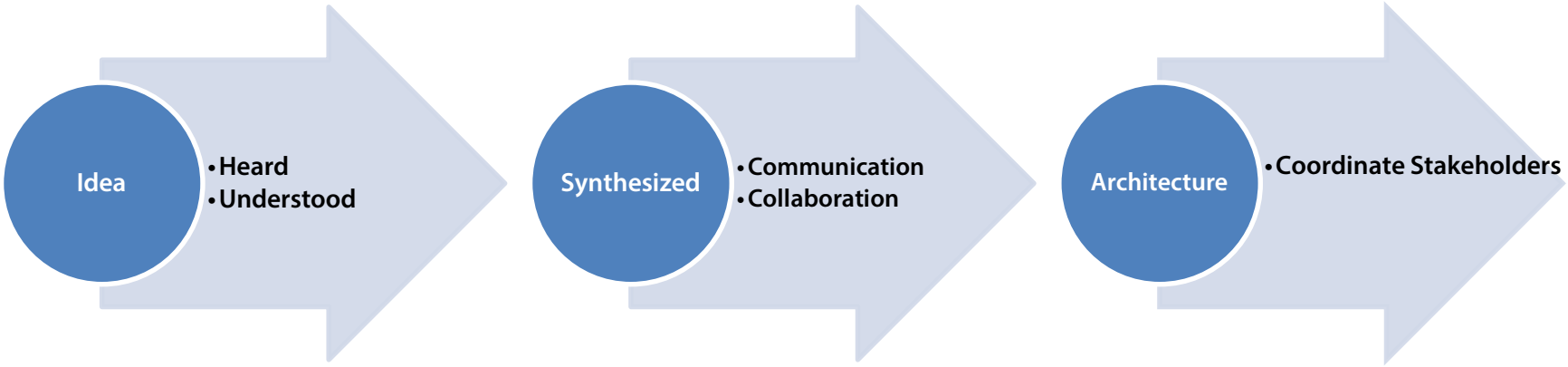
Who is a Software Architect?

"architects work in ways that allow them to accomplish their dual mission of designing technically possible and organizationally feasible products. They bring people from all over the corporation in to consult on their technical knowledge, and at the same time learn about other groups' priorities and schedules. The architects present their work to different groups to ensure that the solutions are attractive to build, buy, and sell. Finally, they continually look outside the company to align their work with standards agencies and competitors. All of this is necessary design work, without it, solutions might work, but could not be built or sold."

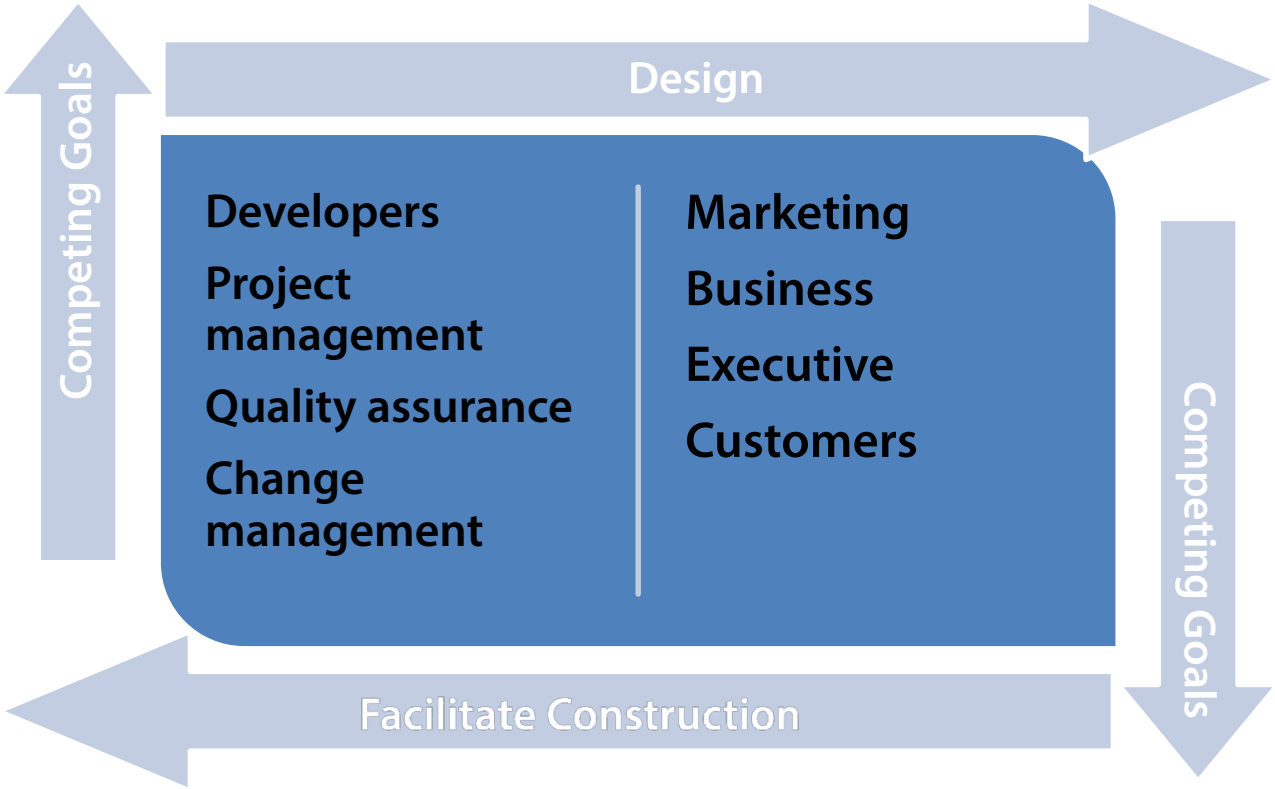
-- Rebecca Grinter –

Who is a Software Architect?

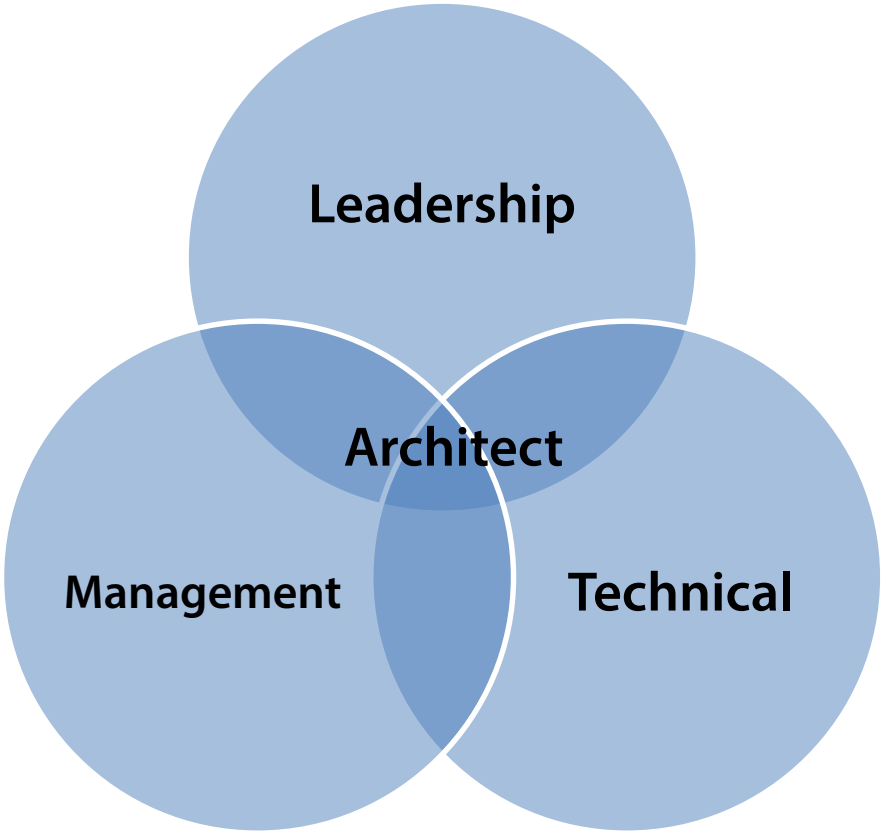
Who is a Software Architect?



Who is a Software Architect?



What is an architect expected to be skilled at?



What is an architect expected to be skilled at?

Technical	Management	Leadership
• Developer • Designer • Modeling • Keeping up new technologies	• Project Management • Problem Solving • Negotiation • Facilitation	• Vision • Self Directed • Decisive • Motivational • Inspirational • Confident • Committed • Delegate • Positive • Creative

Communication & Collaboration

What is an architect expected to be skilled at?

- Books
- Podcasts
- Classes on management & leadership
- Learn from managers in your organization

What is an architect expected to know?

- Broad knowledge
- General list of knowledge areas
- Architect role is very different from senior developer
- New challenges and opportunities to grow
- Senior Developer → deep knowledge specific technologies
- Architect → broad general knowledge

What is an architect expected to know?

Non-Technical

- Leadership Techniques
- Management Techniques
- Business Processes
- Business & Enterprise Domain

Technical

- Architectural Concepts
  - Architectural Patterns
  - Architectural Styles
- Design Patterns
- Software Engineering
- Software Design
- Programming
- Platform Knowledge

What is an architect expected to do?

- **Duties of architect vary with role**
- **Duties overlap**
- **Many Titles**
  - Enterprise
  - Solution
  - Solutions
  - Application
  - Software
  - Infrastructure…
- **Confusing!**

Types of Architects

Enterprise Architect

- Strategic Architect
- People
- Process
- Information Flow
- Business Processes
- Strategic Goals
- Business Strategies
- Technical Strategies
- Roadmaps
- Not all software
- Align with business

Solutions Architect

- Tactical Architect
- Cross Domain
- Cross-functional
- System Interactions
- Frameworks
- Infrastructure
- Interoperability
- Horizontal Problems
  - Roadmaps
  - Guidance
- Cross Cutting
  - Reuse
  - Process
  - Guidance
- Strategic Solutions

Application Architect

- Operational Architect
- Single Application(s)
- Single Technologies
  - Web
  - Desktop
  - Mobile
- Component Reuse
- Maintainability
- Detailed Designs
- Components
- Modules
- Classes
- Libraries
- Languages
- Reuse with business unit

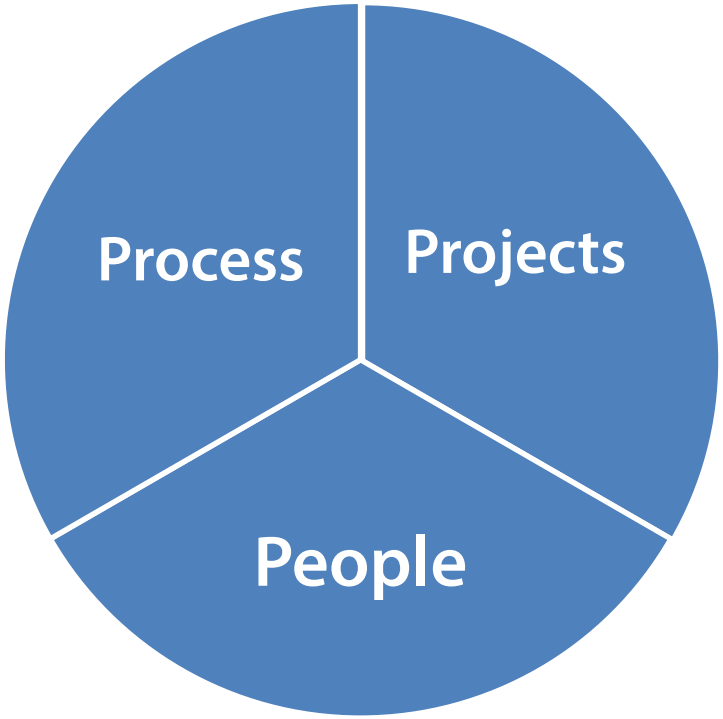
What is an architect expected to do?

- **Perform duties of both Architects**
- **Solutions Architects rarely focus solely on tactical/strategic problems**
- **Solutions Architects deliver operational solutions**
- **Separate Teams for Architects**
  - Solutions Architect Team
  - Applications Architect Team
  - May share same title: Software Architect
- **Two distinct roles:**
  - Solutions Architect
  - Application Architect
  - No matter what your title you may perform the duties of both

What is an architect expected to do?

- Broad set of duties
- Solutions vs. Application Architect?
- Role Varies between organizations
- Core set of duties

Non-Technical Duties – the hard skills



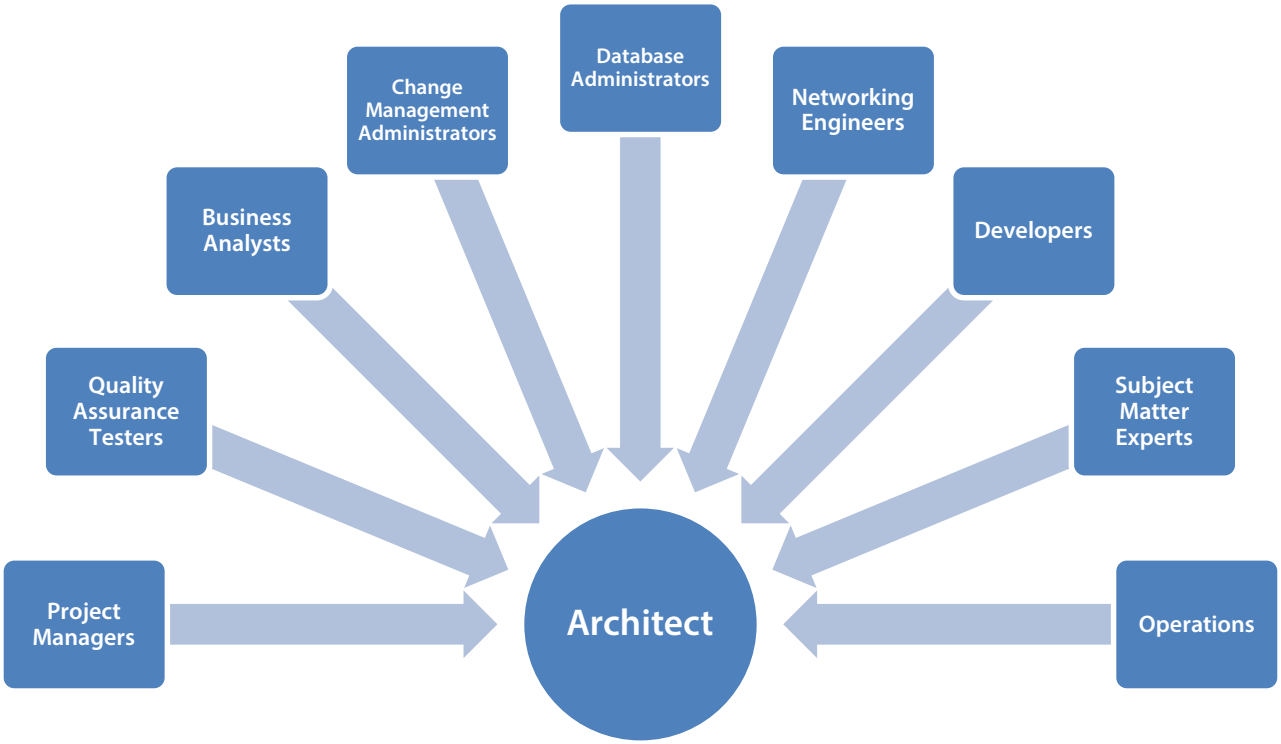
Non-Technical Duties – the hard skills

- **Manage the Project**
  - May have project manager assigned to your project
  - Good project managers are incredible assets
  - Good project managers are hard to find
  - Be prepared to take on the role of the project manager

Non-Technical Duties – the hard skills

- **Build a Team**
  - Required for every project
  - Not only developers on this team

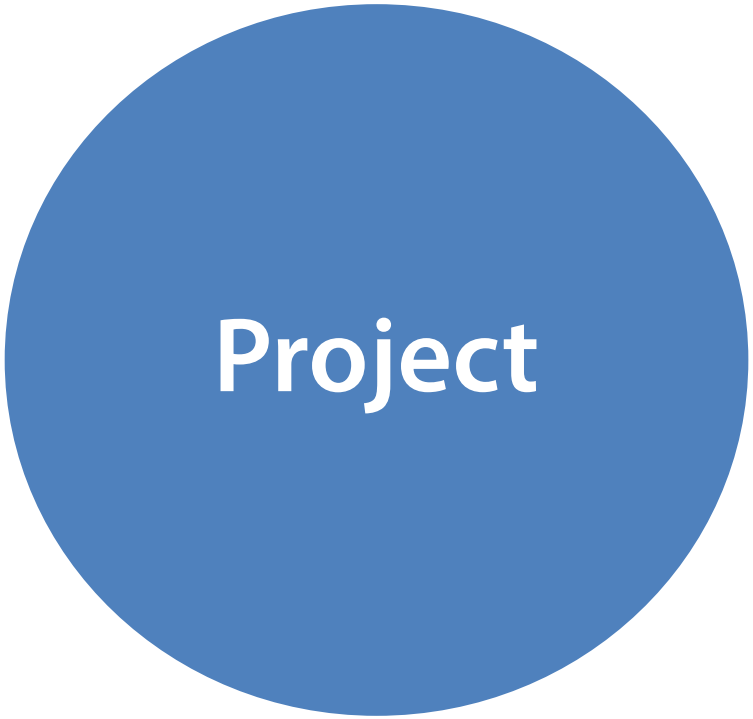
Non-Technical Duties - Team



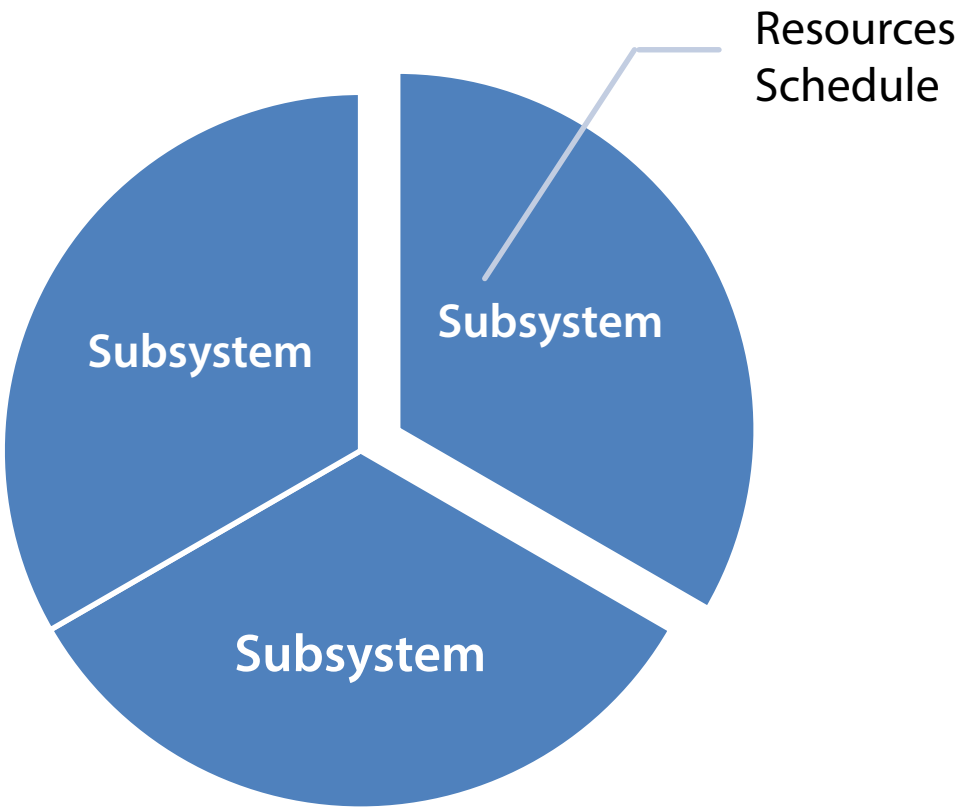
Non-Technical Duties - People

- **Even small projects take a team**
- **Facilitate collaboration between team members**
- **Architects lead the charge**
  - Guidance
  - Technical Decision
- **Project Managers can help**
- **Architects job is to organize:**
  - Solution
  - Team
  - Deliverable Order
  - Break solutions into manageable pieces
- **Sometimes you are the project manager**
- **Organizing & Managing team is critical to delivering project**

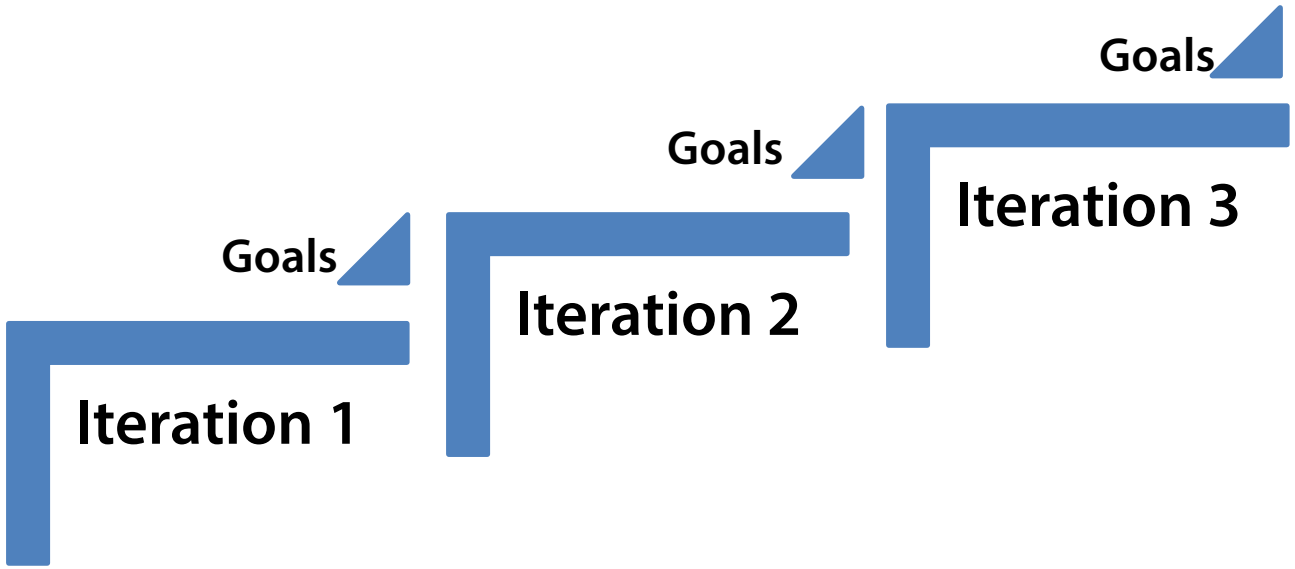
Non-Technical Duties - Project



Non-Technical Duties - Project



Non-Technical Duties - Project



Non-Technical Duties - Project

- **Collaborate with team**
  - Foster ownership
  - Commitment
- **Leading a team**
  - Empower
  - Team sets goals
  - Team sets milestones
  - Architect works with team
- **Checkout the Pluralsight courses on Agile**

Non-Technical Duties - Process

- **Even small projects take a team**
  - Process is your best friend
- **Clearly defined & repeatable process must be adopted**
- **Develop your own processes**
  - Tailor to your organization and methodology
  - Create a set of repeatable procedures

Non-Technical Duties - Process

- **Personal set of documents to guide projects and duties**
  - Simple bulleted lists
  - Continually update
  - Use any tool that allows you to create and organize lists
  - Working set of documents
- **3 Goals**
  - More effective through continual process evaluation and improvement
  - Delegate process
  - Teaching aid
- **Share process documents with team**

Non-Technical Duties - Support

- **Support Management, Business and Organization**
  - Technological Insights
- **Business relies on your:**
  - Expertise
  - Experience
  - Foresight
  - Introduce/Evaluate technical solutions
  - When NOT to apply technology

Non-Technical Duties - Leadership

- **Technical Leader**
  - Technical Team
  - Business
  - Management
  - Organization
- **You are a technical representative**
  - Company, Division, Line of Business
  - Make a good impression
- **Architects provide solutions to the business**
  - Don’t underestimate business creativity
  - Must be open and willing to understand problems
  - Quick to act

Non-Technical Duties - Leadership

- **Technical Leader**
  - Teaching & Mentoring
  - Helps improve colleagues skills
  - Helps project authority and expertise
  - Build relationships
  - Training benefits everyone

Non-Technical Duties - Leadership

- **Leading a Project**
  - Create a Vision
  - Facilitate delivery of solution
  - Delegate
  - Never make yourself the critical path on development tasks
  - Your job is to support and enable the entire team
  - Give junior team members the opportunity to learn & contribute
  - Don’t spread yourself too thin on development tasks

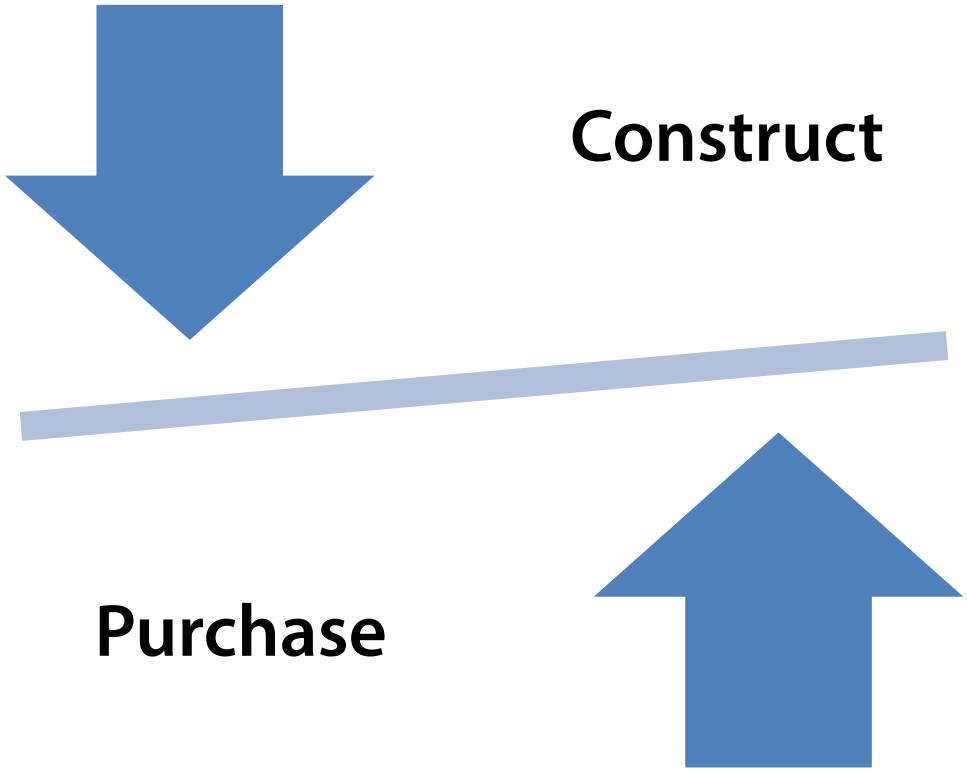
Technical Duties

- **Create an architecture**
- **Small part of the job**
- **Architecting a solution**
  - 3 primary steps

Select the Architecture

Style	Pattern
• Client Server • Message Bus • Service Oriented Architecture • Domain Driven Design • Layered Architecture • Component Based • …	• MVC • Publish/Subscribe • Request/reply • Peer-to-peer • …

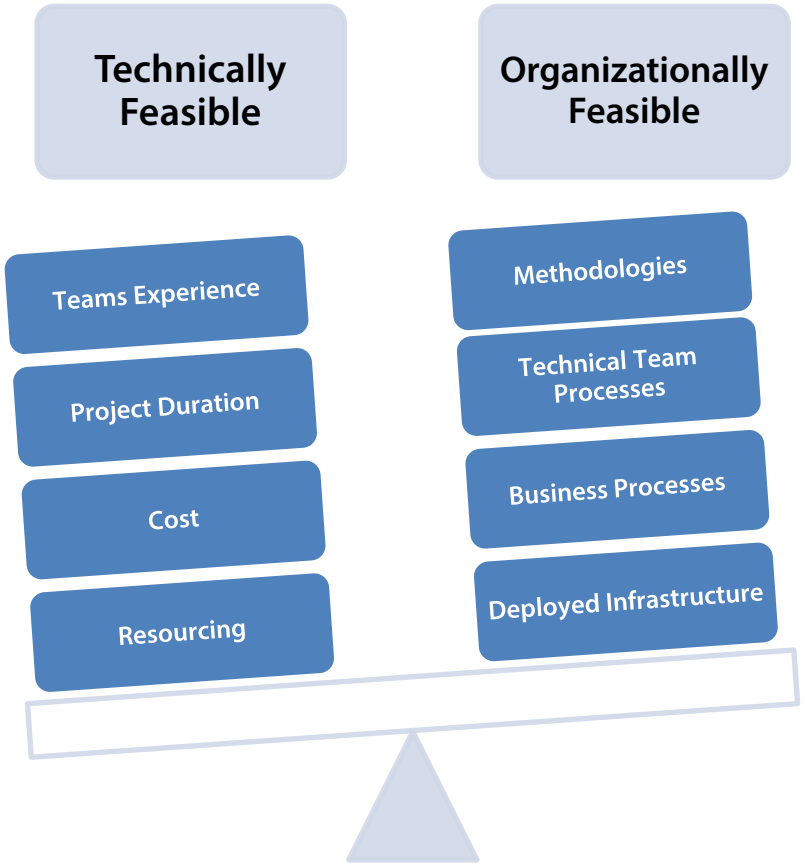
Select the Architecture



Technical Duties

- Design a solution that solves business problem
- Not just drawing rectangles on a white board
- Solving a multifaceted problem

Design the Architecture



Design the Architecture

- **Understand organizational tolerances**
  - Business
  - Technical
- **Design solutions**
  - Solve problem
  - Can be delivered

Document the Architecture

- **Design and Documentation solve two distinct problems**
- **Design – Solve the problem**
  - Understanding entire problem
  - Creating technical solution
- **Documentation – Communicating the solution**
  - Technical & Non-Technical Stakeholders
  - Expression of the design
- **Format and Style are not important**
- **Understanding across the organization is paramount**
- **Goal: Effectively communicate solution**
- **Doesn't have to be UML**
- **Standards are important**

Evaluate Architecture

- **Evaluate your own architecture**
- **Evaluate peers architecture**
- **Evaluating existing architecture**
- **Evaluation**
  - Objective
  - How well solution solves the problem
  - Don’t be overcritical or unconstructive
  - Addresses quality attributes
  - Meets goals and project and organization

Provide Technical Vision, Guidance & Leadership

- **Technology related vision**
- **Technology related guidance**
- **Technology related leadership**
- **Guidance & Leadership when considering solutions**
  - Custom application
  - Purchased products

Evaluate technologies

- Apply technology to business problems
- Keep up with new technology
- Understand existing technology
- Guidance and Leadership – new & existing technology

Select Tools and Technology

- Tools & Technology
  - Development
  - Project & Processes

Code

- Code if you are lucky
- Many non-coding duties
- Up to you how much coding you do
- Very difficult to both lead the team and code
- You assign logical groupings to team
- Can assign pieces to yourself
- Do not make yourself the critical path on any development task
- Your primary duty: Facilitate & Enable team

Code

- Many architects don't code
- Extremely important for Architects to continue coding
- Benefits
  - Understanding new technologies
  - Keeping our existing skills sharp
  - Keep up with new technologies
  - Foster kinship with development team
- How?
  - Training Sessions
  - Prototyping

What an architect should not do

- Ivory Tower Architect
  - Casts down designs from on high
  - Ineffective Technique
  - Makes us all look bad
  - Do not become this type of Architect
- Team effort
- Must be done collaboratively
- Team will provide:
  - Feedback
  - Insight
- Effective Leaders:
  - Work alongside team
  - Listen to feedback
  - Guide Team
  - React to challenges

Why do we need Software Architects?

- **The need for an architect increases with the complexity**
- **The more complex a solution the more likely it could fail**
- **Long drawn out failures more common**
- **Most common reasons projects fail:**
  - Project does not achieve business goals
  - Project takes too long
  - Project costs too much
- **Experienced architects mitigate risk**
- **Experienced architects bring:**
  - Vision
  - Leadership
  - Structure
  - Experience

Why do we need Software Architects?

- **Success can only be planned**
  - Strong Leader
  - Unified vision
  - Careful planning
  - Structure
  - Process

Meeting Business Goals

- **Identify how a solution will achieve its business goals**
  - Project scope and functional requirements
  - Business stakeholders
- **High level architecture presentation**
  - Express the design on paper
  - Explains interpretation of the business goals
  - Collaborative
- **Goals:**
  - Synthesizes all the formal and informal inputs
  - Expresses in a design
  - Communicates
  - Feedback
  - Understand business goals

Meeting Business Goals

- **Collaboration**
  - Listen
  - Speak
  - Guide business toward solution

Project Duration

- **Many dangers when a project takes too long to complete**
  - Business opportunity may disappear
  - Organization may lose interest
  - Other projects may be impacted
  - …
- **Architect helps mitigate this risk**
- **Divide project into subsystems**
  - Each worked independently
  - Abstract complexity
  - Smaller workable chunks
  - Small teams working concurrently
- **Benefits:**
  - More efficient
  - Small wins motivate
  - Hit dates more accurately

Project Duration

- **Realistic time and resourcing estimates**
- **Components are more easily estimated**
  - Top down estimates
  - Bottom up estimates
  - Set realistic expectations with business
- **Determine project feasibility based on top down estimates**
- **Predict project duration based on bottom up estimates**

Project Cost

- **Architects have much influence on cost**
- **Architects quantify costs of components**
  - Understand which are more complex & risk prone
  - Guide business to less costly & less complex solution
- **Business can't quantify cost**
  - Need vs. Want
- **Architect helps business understand needs with regard to cost**
- **Architect should help understand cost associated with project**
  - Provide alternatives
  - Same goal
  - Lower costs
  - Reduced complexity
- **Focus on what they are trying to accomplish?**

Project Cost

- **Always ask why**
  - Listen
  - Understand
- **Needs and wants are often combined**
  - Asserted as facts
- **Whole process is a negotiation**
- **Job of the Architect:**
  - Listen
  - Identify needs
  - Guide
- **Not easy but necessary to keep projects within budget**

Righting the Ship

- **Architects help right the ship**
- **Many reasons why project veer off course**
  - Too long to complete
  - Cost more than anticipated
- **Reduce time and cost**
  - Reduce scope
  - Identify complex components
  - Eliminate or move complex components
  - Effective resource management
- **More developers doesn't always help**
  - As developers increase so does management and ramp up time
  - Architect can help identify where most effective to add resources

Why do we need Software Architects?

- **Software architects are a valuable asset to organization**
- **Provided tangible examples:**
  - Project Duration
  - Project Cost
  - Business Goals

Summary

- Software Architect role is very different from that of a senior developer
- Being skilled technically is only 1/3rd of the equation

- Who is a software architect
- What a software architect is expected to be skilled at
- What a software architect is expected to know
- What a software architect is expected to do in the organization
- Why organizations need software architects
- How a software architect might help a project that has veered off course