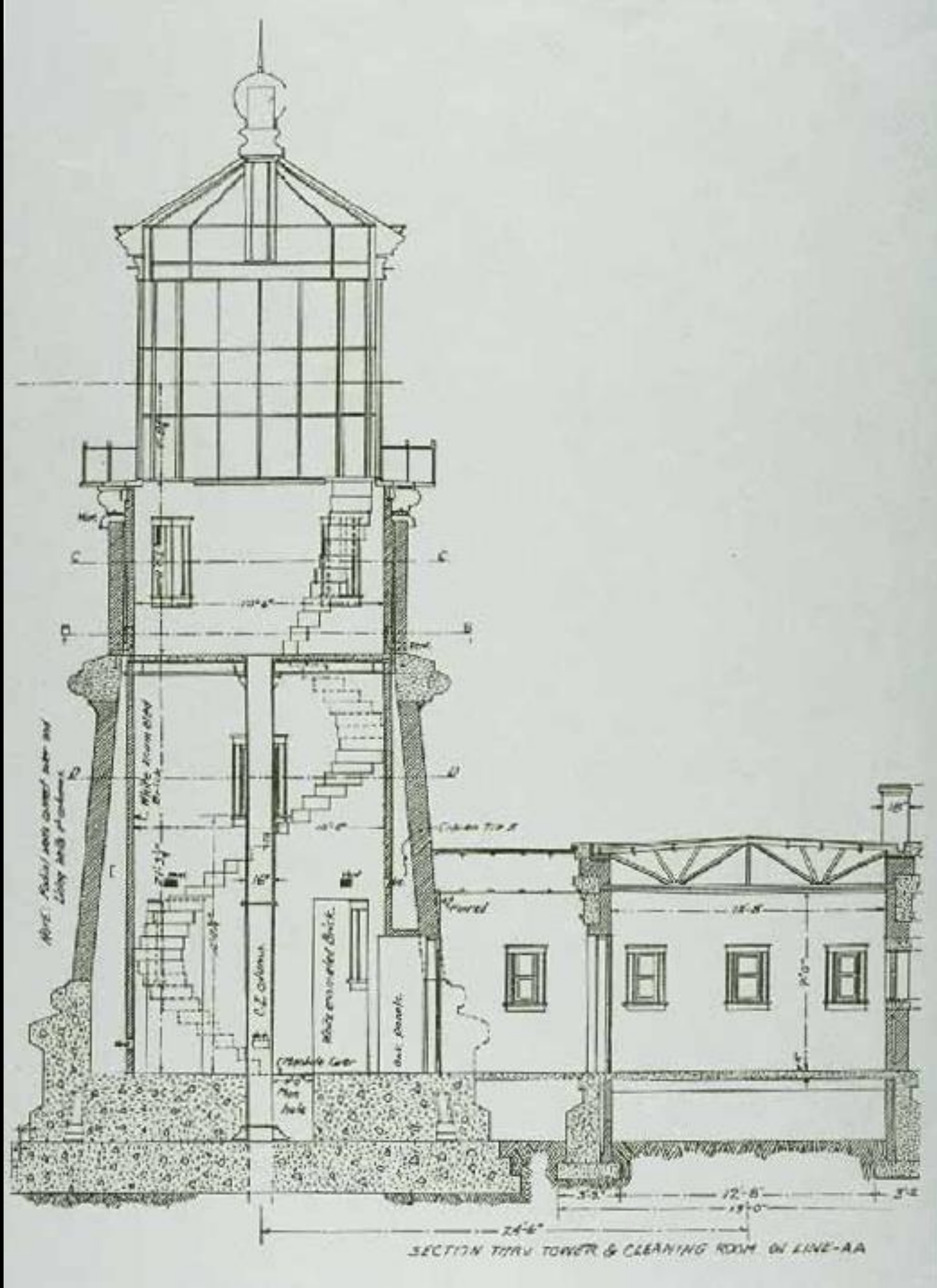
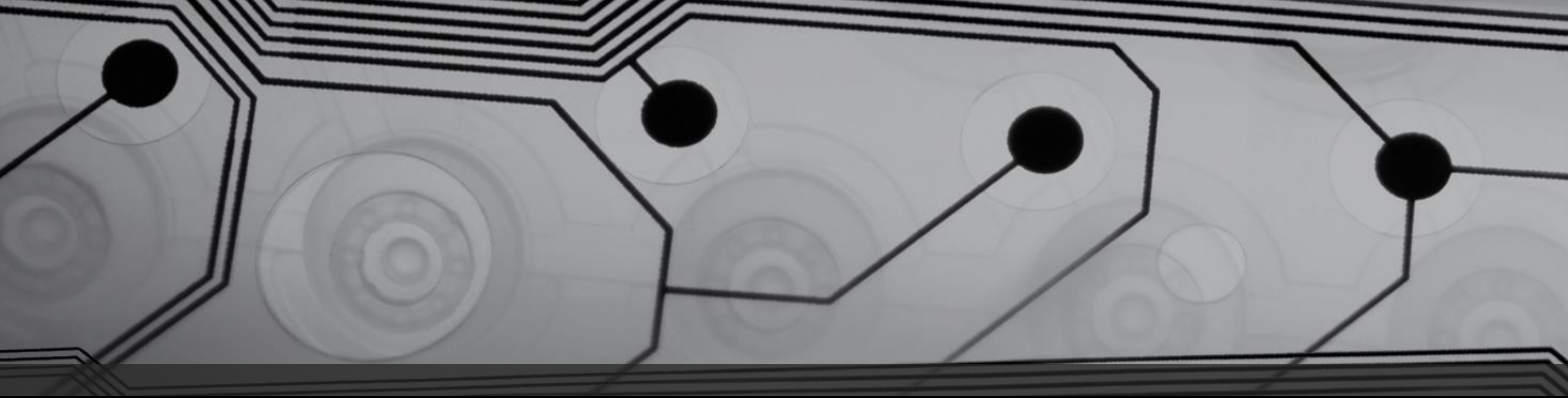
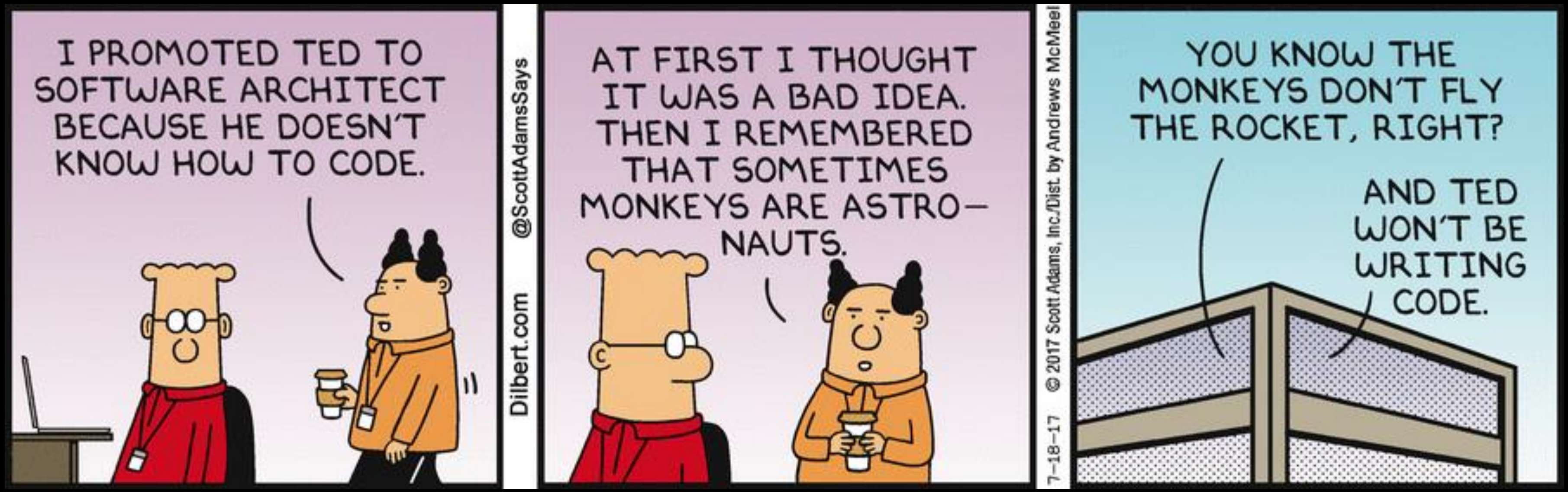
Software Architecture & Design





Role of the Architect

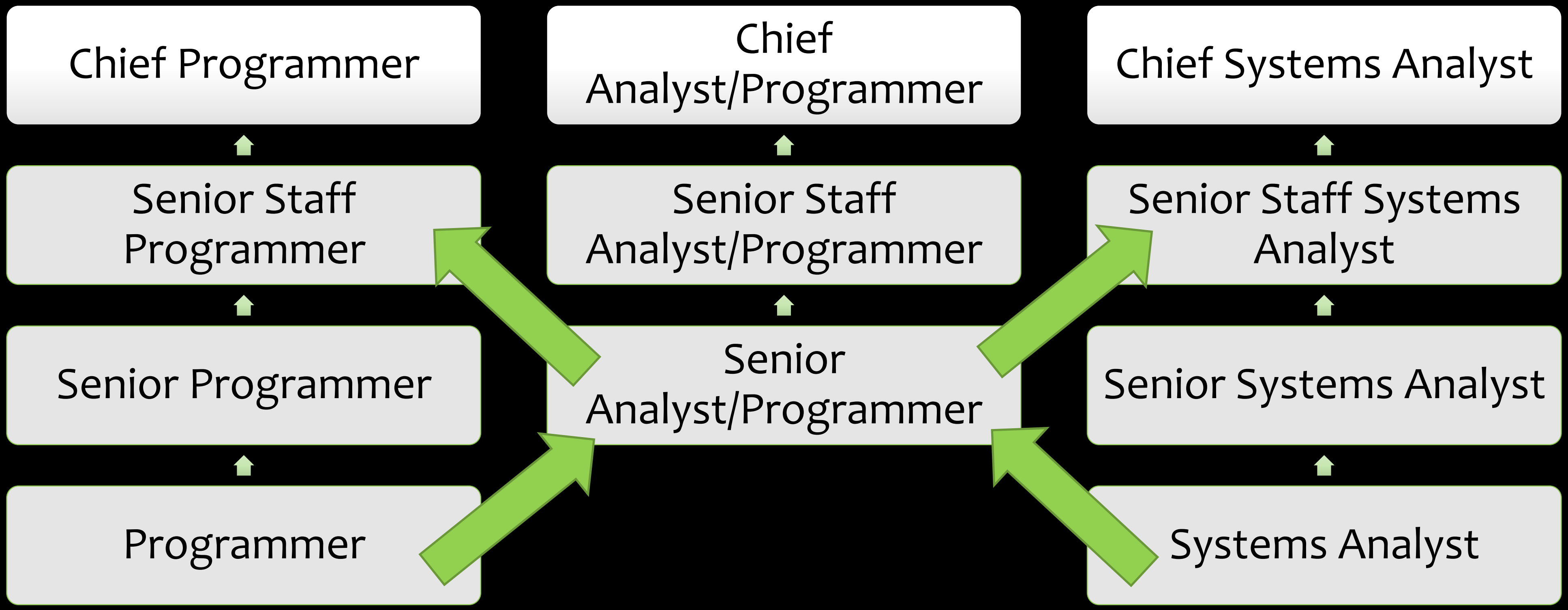
The emergence of the role of Software Architect



I PROMOTED TED TO SOFTWARE ARCHITECT BECAUSE HE DOESN'T KNOW HOW TO CODE.
@ScottAdamsSays
Dilbert.com
AT FIRST I THOUGHT IT WAS A BAD IDEA. THEN I REMEMBERED THAT SOMETIMES MONKEYS ARE ASTRONAUTS.
© 2017 Scott Adams, Inc./Dist. by Andrews McMeel
7-18-17
YOU KNOW THE MONKEYS DON'T FLY THE ROCKET, RIGHT?
AND TED WON'T BE WRITING CODE.

Source: http://dilbert.com/strip/2017-07-18

Roles circa 1990



Chief Programmer

Senior Staff Programmer

Senior Programmer

Programmer

Chief Analyst/Programmer

Senior Staff Analyst/Programmer

Senior Analyst/Programmer

Chief Systems Analyst

Senior Staff Systems Analyst

Senior Systems Analyst

Systems Analyst

Narrow Range of Technologies

Programmer

- COBOL, PL/1, Assembler
- DB2/SQL/CICS

Analyst

- LSDM (Methodology)
- Entity Modelling

Shift to a greater set of Technologies

Programmer

- Visual Basic
- Visual C++ & MFC
- UI Design (Windows)
- ODBC
- T-SQL & SQL-Server
- Client/Server

Analyst

- UML
- OO Analysis
- OO Design
- Prototyping & Rapid Application Development (RAD)

Shifting from C to C++

C

- Macros
- Pointers
- Structs
- Arrays
- Typedefs
- Functions

C++

- Macros
- Pointers
- Structs
- Arrays
- Typedefs
- Functions

- Classes
- Inheritance
- Private & Protected Members
- Namespace
- Function Overloading
- Constructors
- Destructors

- Default Params
- Inline Functions
- References
- Const
- Friends
- Templates
- Exceptions

Technology Shifts in early ‘90s

- Object Oriented Programming
  - Lead to a mix of analysis & design
  - Combining function and data – forced to think about design
  - OOP facilitated/encouraged more complex design, leading to more complicated applications
- WIMP & WYSIWYG
  - Windows, Icons, Mouse & Pointers
  - Vastly richer UI – and hence more complicated
- PCs & Networks
  - Client/Server - involved the network
  - Concurrency Control

The Result?

- Greater level of Complexity
- Increased Levels of Abstraction
- Drive for re-use – the promise of OO
- Need for increased degree of System Oversight

Emergence of the Architect

- Focus on the high-level design choices – i.e. The ‘Big Picture’.
- Chose technologies/technology stacks & understand their interaction → Consistency
- Determine the projects technical ‘philosophy’ & control/limit Design Choices
- Define the ‘component model’ – i.e. high-level object model
- Design for Re-use; Design for change; Design for extensibility (insulating layers)
- Determine Coding Standards
- Communicate these concepts to the developers/engineering teams

Types of Architect

Role	Description
Solutions Architect*	• Typically focused on a BU • Concerned with interaction between applications • Plugging systems together to form a solution • Communicates with several teams
Application/Technical Architect	• Focused on application development – i.e. product • Concerned with internal design, component reuse, code quality, maintainability & extensibility • Communicates across the engineering team • Often a strong developer/engineer
Embedded Systems Architect	• Very focused on the hardware & embedded system • Concerned with embedded system internals • Communicates within team & interacts with Application/Solution Architect.
Enterprise Architect	• Interaction between BUs and IT • Business Transformation • Scope is across the organization • Highly abstracted design • Communicates across the organization

Challenges – Not recognizing the politics of architecture

- Architecture is a political act
- Engineers like to believe they are outside politics
  - They are not – where you have humans, you have politics
- The architect is arguably the most prestigious role in software development
- Dangerous assumptions:
  - All developers have honest intentions with regard to the project
  - We want to believe the best course of action is the obvious one (and everyone will agree)
- No one likes confrontation

Pitfalls of Object-Oriented Development *by Bruce Webster*

Architect Needs

- Authority
  - Authority commensurate with Responsibility
  - Authority to compel adherence
- Upper Management Support
  - Full backing of management hierarchy
- Proven Track Record
- Pact with Engineers that serious consideration will be given to all ideas & suggestions
- Willingness to give ground on smaller design issues to ensure integrity of the bigger, more important design decisions

Challenges – Don’t expect thanks

- Much of Software Architecture is invisible
- Generally, no one says ‘good job’ for ‘*ilities’:
  - Scalability (+ Performance)
  - Recoverability
  - Extensibility
  - Adaptability
  - Performance
  - Robustness

Skills

- Skills Required of the Architect
  - Hard Technical Skills
  - Fuzzier Analysis & Design Skills
  - Soft skills

- Perspective
  - Set the vision
  - Ability to zoom out and see the ‘bigger picture’
  - Ability to zoom in to the lowest detail*

- Communication
  - Deal with engineering management & upper management
  - Might deal with C-level executives, like CTO
  - Communicate to all levels of engineering
  - Communicate with DevOps

*Controversial

Tomcat
Maven
SaaS
MariaDB
Storm
CloudFoundry
Kafka
Javascript
Shell
PaaS
IntelliJ
SpringXD
Java
Cloud
Git
Hadoop
MySQL
Docker
HBase
IaaS
Microservices
Chef
SpringBoot
AngularJS
SQL
HAWQ
MongoDB
Spring
Data Analytics
MQTT
Puppet
Distributed Systems
AMQP

Assertiveness

Team Communications

Essential Leadership Skills

Influencing Skills

Presentation Skills

Strategic Decision Making

Time Management

Technical Writing

Emotional Intelligence

Interviewing Techniques

Negotiation Skills

Q&A

Discussion Time

Summary

- Role of Architect is now firmly established
- Complexity of modern software development means someone (or some team) needs to manage the ‘Big Picture’
- Can mean different roles, depending on organization - There are different types of architect.
- Challenging Role – Need to be a master of the ‘soft skills’
- Needs Upper Management Support – Needs a degree of Authority

Thank you

Recommended Reading

- https://en.wikipedia.org/wiki/Software_architect
- https://www.sei.cmu.edu/architecture/research/previousresearch/duties.cfm
- http://www.codingthearchitecture.com/presentations/dw2009-pitfalls-for-new-software-architects/