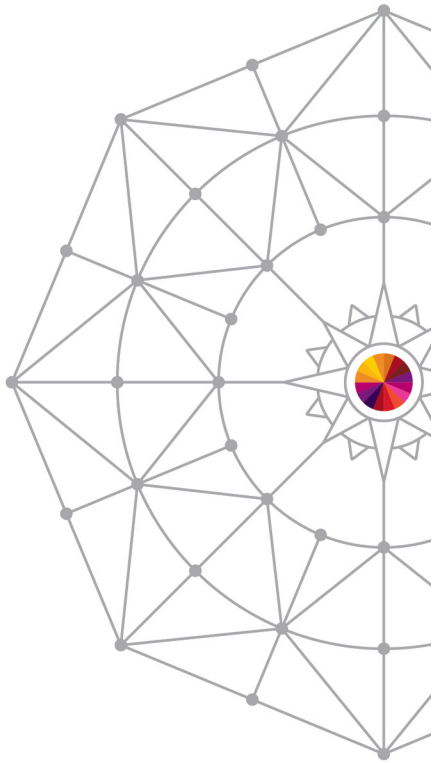
So You Want to be a Software Architect?

Session #15418

Ellis Holman
IBM Client Technical Architect
Phone: 317-249-9551
eMail: eaholma@us.ibm.com



#SHAREorg

Disclaimers for lawyers

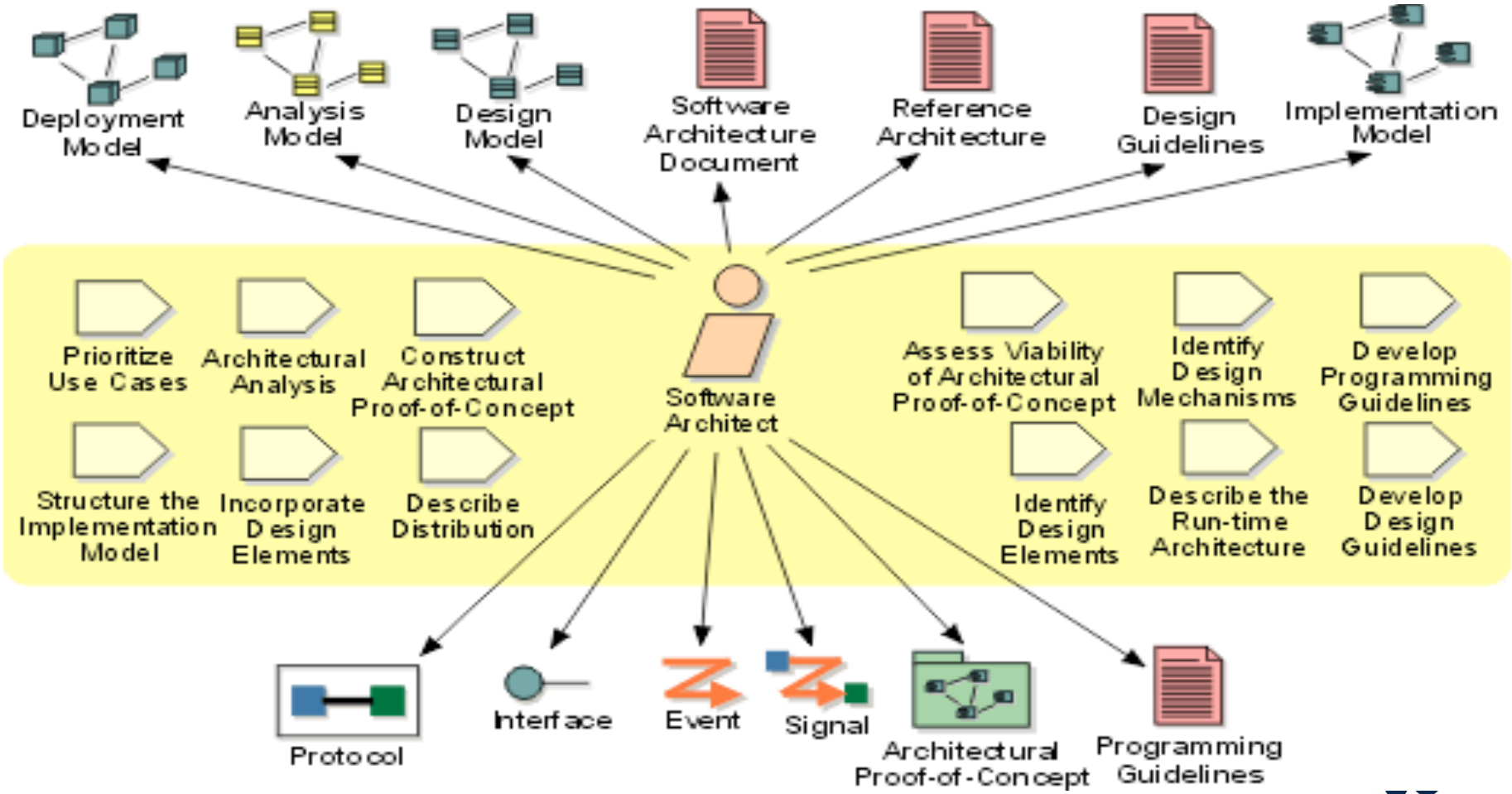
- Websites are in a constant state of change, where URLs are given they were active at the time of review
- All trademarks are the property of their respective owners
- No judgment of suitability or appropriateness of use is implied
- No recommendation or promotion of products mentioned in this presentation is implied

Software Architect
is
Someone Who Can
Make
Sub-optimal Decision
in
Total Darkness

Where did Software Architecture come from

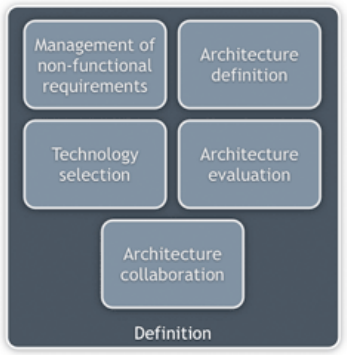
- Driven by business need
  - Cost cutting
  - Increase Revenue
  - Strategic Advantage
- Driven by new technology
  - Main Frame/Batch
  - PC/interactive Real time
  - LAN/WAN
  - Internet
- Driven by Software Engineering principle, methodology
  - Functional decomposition, sub routines

An architect will be many things to many people



So what is a Software Architect?

- Some disagreement as to what exactly a Software Architect is
- Some basic terms associated with the title:
  - A computer manager who makes high-level design choices and dictates technical standards
  - Software architecture is all about having a view of an entire system and seeing the individual parts fit to understand how the software system works as a whole



So what is a Software Architect?

- A software architect is really an expert position
  - Has responsibility for selecting the components and interaction patterns used across a whole project to achieve that project's goals
  - If the architect gets it wrong, the project will either fail completely (with huge recriminations on all sides)
  - Succeeding? with results that can't be rolled into production; because, they're so complex only the original developers can understand the design
  - Software architecture has to bear in mind project management too

An Architect is the sum of their experience

- Software architect generally requires a bachelor's degree in a computer discipline and additional training or credentials
- Continuously seek to improve
  - If there was one right way to do things, the role of an architect would not be needed
  - One obvious way to improve in the area of architecture is to read
  - All platforms, all technologies, not just your comfort zone
- Learn a new programming language every one to two years.
- Focus on an area
  - Have a high-level understanding of as many technologies as possible

An Architect is the sum of their experience

- Play with different technologies, design patterns, architectures, etc
- Learn to speak the "language" of your target audience
  - You have to speak to a lot of different people as an architect
  - Each audience will have a different level of understanding of technology.
  - Learn to tailor your explanation in ways that each audience can understand
- Read magazines, go to user group meetings and technology conferences like SHARE
- Speak at SHARE and other conferences
  - Helps build knowledge
- Discipline is key
  - Always do your best work, even if it doesn't sound like the most fun.
  - Schedule time every day to learn something new
  - Don't let other priorities take over this time.

Personal qualities required to be a software architect

- Software engineers need strong technical, analytical and problem-solving skills
- Developing software from conception to completion requires creativity, as well
- Excellent customer service and communications skills are also needed
- Software architects work with non-technical business managers and employees, and must translate technical jargon into terms users of the software understand
  - I call this the ability to speak English
- A critical role (but not the only one) is the ability to understand users' needs

Establishing patterns, becoming an ‘expert’

- It takes experience to become an architect
- Not be counted in the number of years, but in the amount of wisdom gained from it
  - There's no real short cut to that
  - It relates to how long your brain takes to lay down certain types of pattern
  - Understand the problems in terms of multiple levels of abstraction
  - That's not very easy, because each level of abstraction has its own restrictions and domain of applicability
  - Finding a good balance, especially one that you can explain to others, is difficult

Earning your ‘creds’

- You can make an excellent impression in one meeting,
  - Gaining the standing and credibility that makes others rely comfortably on you in important matters takes many years
  - A senior enterprise architect is in a position where C-level executives and board members must sometimes bet their careers on the correctness of their judgment,
  - Errors can result in millions of dollar misspent
  - Excellent track record is best earned over a few years, in several different positions, with extensive experience in a wide-ranging array of topics and technologies
- Part of your credibility is the respect you enjoy with your peers and in the larger community
- If you’re a respected and a top contributor in one or two fields, this might be beneficial to your credibility
- If you’re a respected member of the developers community at your employer and colleagues tend to seek your advice, this might persuade others to listen to you more carefully
- Build the standing required to push through unpopular positions so that your word becomes of actual value to the C level types, because it was in the past
- It all boils down to: providing good value to many people over years

Start with ‘why’

- We want to make sure that an architectural solution will answer a real business scenario within the system
- Do we need to provide fault-tolerance or 99.999% uptime
  - What is the business requirement
  - We need to define why we need high-availability
  - What will be the consequences of the downtime loss of a business transaction?
    - Loss of 100K dollars? loss of life
    - It can save you money
  - It will help you get to the right requirements
  - It will help you justify your decisions
- Critical thinking is not just a good idea, it is a requirement

Will you be able to answer these ‘simple’ questions

- Is this a “Good Idea”?
  - Feature creep — kill it or follow-on work
- DRY? Do I repeat this anywhere?
- Orthogonal?
  - How independent is this?
- Testable? How will I test this?
- Is there another way?
  - Having only one idea is dangerous
- Costs of changing
- What would the architecture look like if I didn't have this problem?
- What are the facts and assumptions?
- **Document rationale**

Questions?

Sources

- “Software Architecture for Developers”, Simon Brown
- Barbacci, Mario, " Are Software Architects Like Building Architects?" *SEI Interactive*. http://interactive.sei.cmu.edu/news@sei/columns/the_architect/1998/September/architect-sep98.pdf
- Rechtin, E. *Systems Architecting: Creating and Building Complex Systems*. Prentice-Hall, 1991
- Worldwide Institute of Software Architects (WWISA) http://www.wwisa.org/wwisamain/index.htm

Sources (continued)

- Bredemeyer, Dana. "James Madison and the Role of the Architect", http://www.bredemeyer.com/madison.htm, 1999.
- Bredemeyer, Dana and Ruth Malan. " Role of the Software Architect", http://www.bredemeyer.com/role.pdf, 1999.