What an Agile Architect Can Learn From a Hurricane Meteorologist

Eric Richardson
Architect for Editorial Systems
Agenda

• Overview of CAS
• The analogy and some terminology
• Predicting the future
• How to get there from here
• The cone of uncertainty
• Hurricane hunters
• Influences on and impediments to architecture
Overview of CAS
CAS is a division of the American Chemical Society

ACS Mission

To advance the broader chemistry enterprise and its practitioners for the benefit of Earth and its people.

CAS Mission

To provide the world’s best digital research environment to search, retrieve, analyze, and link chemical information.
CAS is the world’s authority for chemical information

- Headquartered in Columbus, Ohio, hundreds of CAS scientists worldwide analyze and index publicly disclosed scientific information

- CAS databases include content from 10,000 major scientific journals, 61 patent authorities, dissertations, meeting abstracts, electronic-only journals, and more. They are updated daily, curated, and quality-controlled by CAS scientists

- CAS maintains CAS REGISTRY℠, the “gold standard” for substance information, the largest collection in the world and the only integrated, comprehensive source of chemical information from a full range of disclosed material, including patents, journals, and reputable web sources

- Customers include more than 1,800 universities, top Fortune 500 corporations, and all major patent offices
CAS chemists analyze and index scientific information to ensure our databases are authoritative.
IT specialists create software to support delivery of our databases and the products used to search them
The analogy
Why meteorology and architecture?

- Weather and software systems are similarly complex
- Meteorologists have been dealing with their uncertainty far longer
- Parts of both systems interact in surprising ways
- In both fields, accurate predictions are nearly impossible
Acronym review

- BDUF – Big Design Up Front
- NDUF – No Design Up Front
- NIMBY – Not In My Back Yard
- RPC – Remote Procedure Call
- TfAA – Technique for the Agile Architect
Technique for the Agile Architect (TfAA)

- **technique** – [tek-neek] *noun*. The manner and ability with which an artist, writer, dancer, athlete, or the like employs the technical skills of a particular art or field of endeavor (http://dictionary.reference.com/browse/technique)

- Derived from fundamental principles
- How you use technique will depend on the context
- Best mastered with practice
Predicting the future
The false dichotomy of Agile Development and Architecture

• **Architecture promotes**
  — Desirable qualities (stability, reliability, etc.)
  — Predictability
  — Long-term “goodness”

• **Agile development favors**
  — Individuals and interactions over processes and tools
  — Working software over comprehensive documentation
  — Customer collaboration over contract negotiation
  — Responding to change over following a plan

*From the “Manifesto for Agile Software Development” http://agilemanifesto.org/*
Roots of the tension between Agile and Architecture

- Architects find it hard to resist BDUF
- Agile practitioners scorn prior planning
- Architects hate to be wrong
- Agile teams need to get something done

http://www.enagility.com/comics/agile-comics/
Adding the Agile to Architecture

- Analysis and design have to change
- Comprehensive approach takes too long
- Short-term accuracy is critical
- Accuracy becomes less important over time
- Predicting business needs beyond the planning horizon is futile
TfAA: Make a forecast

- Treat your architecture as a forecast (desired future state)
- It is going to be wrong!
- BDUF* is an anti-pattern; so is NDUF**
- The team needs the architecture before they start

*BDUF – Big Design Up Front
**NDUF – No Design Up Front
Timeliness in spite of incomplete information

• Agile incompatible with BDUF
  — Working software over comprehensive documentation
  — Responding to change over following a plan

• Agile also does not mean NDUF
  — Wasted effort
  — Missed opportunities

From the “Manifesto for Agile Software Development” http://agilemanifesto.org/
Agile Architect’s responsibilities

• Change the forecast; it is “wrong” anyway

• Your forecast
  — Has to be accurate for the near term
  — Does not have to be completely accurate for the long term
The cone of uncertainty
Agile Architecture has to be mutable

- Your system architecture is what it is
- Achieving a desired state will require change
- Decide how to get from where you are to where you want to be
Evolving architectures have some uncertainty

- **RPC has evolved**
  - CORBA, DCOM, RMI, .NET
- **Web services have evolved**
  - XML-RPC, SOAP, REST
- **Service hosting has evolved**
  - Data Center, Collocation, Hosting, Cloud
- **Business climate changes**
- **What is not a viable alternative today, may be imperative tomorrow**
- **The architect has accumulated more experience**
TfAA: Create a projected path

• Decide what changes to make
• Decide what to change first
• Remember:
  — The target will move
  — Changes should take you closer
  — Being a little wrong is OK
TfAA: Create a projected path

A projected path

- Is flexible
- Is a plan for evolution
- Is your hedge against surprises
- Addresses the risk of revolutionary change
Hurricane hunters
Architects need help

• Architects have to know what is being implemented
  — How is the plan working?

• An architect should “code dive”… occasionally
  — Ability and willingness are both essential

• Getting help can be more effective than DIY
A word about code diving

• Great way to learn about
  — New technology introduced to solve a problem
  — Changes implemented and integrated by best and brightest
  — Suitability and tradeoffs of changes

• Takes valuable time
Knowledge of implementations helps architects

• Evaluate understanding of the plan
  — How effectively did we share the architecture?
• Evaluate fitness of the plan
• Find gaps in the analysis
• Determine if the plan is being ignored
  — Could be the fault of the plan
  — Could be other things
TfAA: Identify and recruit hurricane hunters

- By creating a team of hurricane hunters, the Agile Architect
  - Lets the people best suited to the job do the job
  - Builds relationships with leaders in the trenches
  - Identifies and recognizes the top performers

- Hurricane hunters have their own skills and priorities
**TfAA: Identify and recruit hurricane hunters**

- **Business analysts**
  - Know and understand the business goals

- **Technical people close to the implementations**
  - Know and understand the architectural principles

- **Experienced people**

- **Trust is a two-way street**
  - Architect has to trust the hunter
  - Hunter has to be comfortable talking to the architect
TfAA: Build a network of weather stations

- Implementers are sometimes the first to see the forces of change
  - Developers with software
  - Sales and marketing with business

- Information will be filtered and aggregated

- An architect needs this information
TfAA: Build a network of weather stations

• Tap into communication channels
  — From the Architect
  — To the Architect

• Access much more specific information

• Use the collective knowledge
  — The network knows more than the station
TfAA: Monitor and adjust

- Nothing is as constant as change
- Yesterday’s decisions may be wrong today
- Wrong today doesn’t mean wrong yesterday
- Frequency and amplitude of adjustment are important
  - Too frequent leads to thrashing
  - Too severe leads to wasted effort
TfAA: Monitor and adjust

- Aggregate what you learn
- Adjust your forecast
- Change your projected path
- Rinse and repeat
Influences and impediments to architecture
Most architectures must change

- Hurricanes are guided by steering winds
- Architecture also has change drivers
  - Business needs
  - Technology
  - Experience
- Predictions are easy when winds are strong
Impediments to architectural change are to be expected

- Hurricanes are limited by shearing winds
- Architecture also has change limiters
  - Culture
  - Legacy systems
  - Technical inertia
Architectural change limiter: Culture

• Layers of bureaucracy for approval to do anything
• Extreme risk aversion
• NIMBY
• No confidence in sustainability
• Remember, there are reasons behind the culture
Architectural change limiter: Technical inertia

• Application silos

• “The way we have always done it” mentality
  - All processes are batch driven
  - Pick up the data from A, manipulate it, drop it into B
  - “I can do that in one line of Perl!”

• Significant investment in a particular technology
TfAA: Cultivate ongoing consensus on major architectural decisions

- Architects can influence forces of change
- Engage the leaders
  - Influence culture
  - Redirect inertia
- Disagreement is inevitable – and desired
  - The intractable will at least understand even if they don’t agree
  - Wind shear is what moderates the destructive power of hurricanes
TfAA: Cultivate ongoing consensus on major architectural decisions

- Understand the business drivers
- Educate and inform the implementers of the architecture
- Listen to dissenting voices
  - To understand what is motivating the dissent
- Be aware, even if you can do nothing about it
Summary

• Treat your architecture as a forecast (a projected path)
• Scope out the cone of uncertainty
• Recruit hurricane hunters
  — Build a network of weather stations
  — Monitor and adjust as needed
• Cultivate ongoing consensus on major architectural issues
Thank You!

eric.richardson@cas.org