Not so clever

how we built a successful risk system guided by simplicity.

Zohar Melamed - Qube Research and Technologies

Setting the scene

- This talk is not really about how to go about building a risk systems.
- The initial group of developers have been working together on and off for quite a while
- We have never built a risk system
- We did however have a very clear set of values which governed the choices we made.

Looking at existing systems

- How easy is it to test the system as closely as possible to how it will run in production
- How clear is the demarcation of responsibility between the various parts of the system
- How frequently can the system be deployed
- How steep is the learning curve

A long standing value system

- We value simplicity
- We avoid magic - more on that later
- We promote technical choice (technically speaking)
- We are firmly in the evolutionary camp of system design/architecture

When applied to a risk system

- Universal addressability
- The risk system knows little about risk - often true for the developers
- Everything is immutable
- Everything is the same - Everything is a Resource regardless of type
- Clarity and visibility is essential for both testing and runtime sanity

Magical mystery tour

- Things that have strong magic - J2EE , Spring , SecDB
- The magician behind the curtain make the complex stuff work
- The people developing the simpler things jut rely on that magic being there
- Removes agency from the people doing the work
- Magical things are often painful to test

Testing as an evolutionary driver

- We spend little time pre-architecting things and even less time debating which tech to use
- We expect to make choices that will not work forever
- We do not know what the system will look like in many respects
- Testability is driving many of the choices we make

Successful software evolves fast

- Premature architecture is an evolutionary constraint
- Monolith to services when change stabilises
- Start with the simple solution and evolve, you will find out the right thing
- Anything that slows down iteration is a target - inside and outside the project

Value system compatible technologies

- http
- URL/URI
- json
- browser

What does it boil down to

- Could it be just about evolution vs design?
- Everything as a means of removing evolutionary constraints
- What do quants really want - Love sophistication and complexity
- What do developers really want - Love surviving being woken up at 3AM with an outage