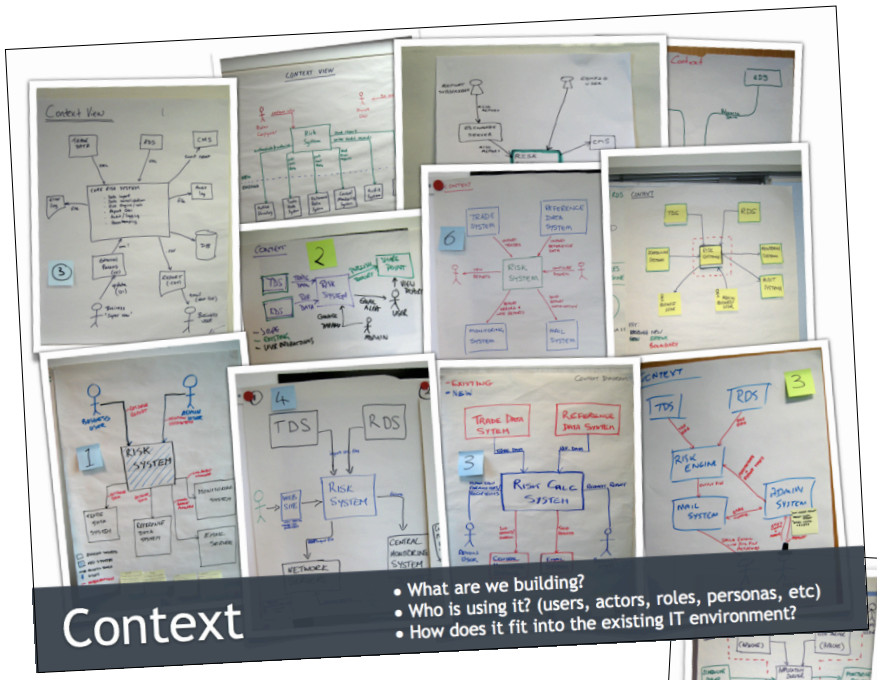
Effective software architecture sketches

1 Agree on a simple abstraction that the whole team can use to communicate, for example:

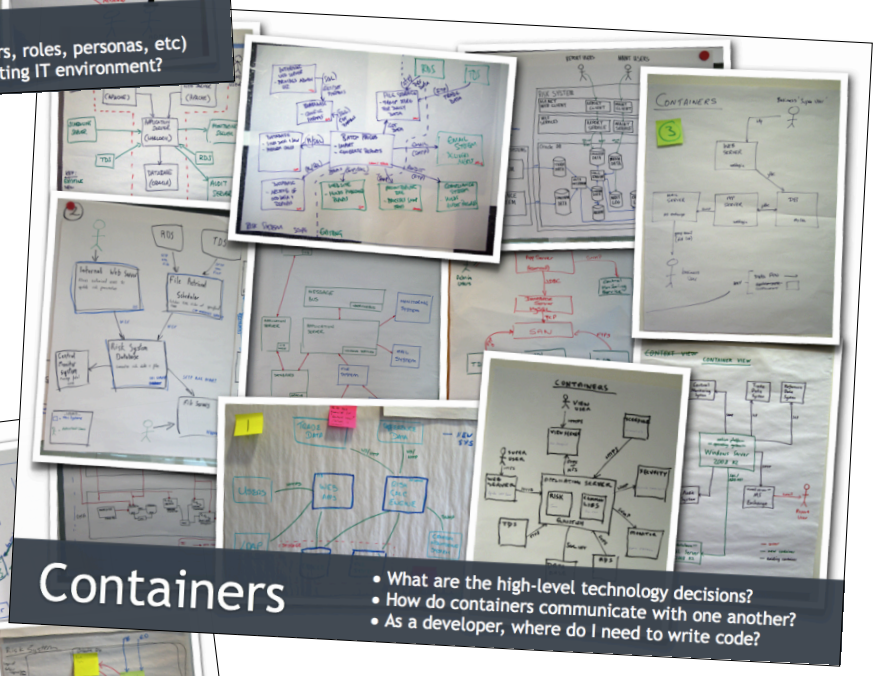
Software System -> Containers -> Components -> Classes

2 Draw simple sketches at each level of abstraction.



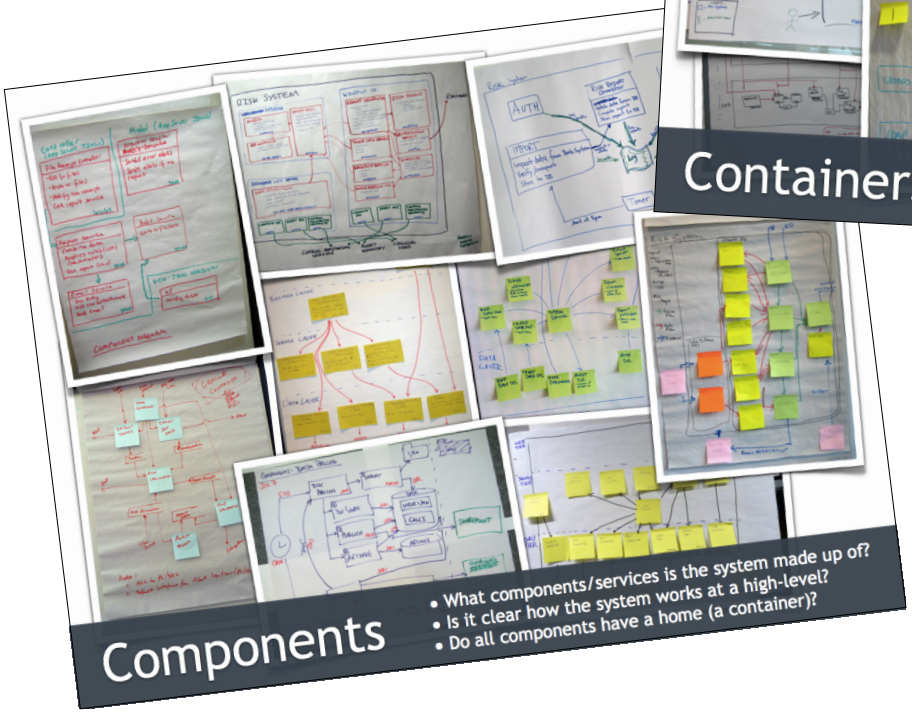
Context

- What are we building?
- Who is using it? (users, actors, roles, personas, etc)
- How does it fit into the existing IT environment?



Containers

- What are the high-level technology decisions?
- How do containers communicate with one another?
- As a developer, where do I need to write code?



Components

- What components/services is the system made up of?
- Is it clear how the system works at a high-level?
- Do all components have a home (a container)?

coding (the) architecture

www.codingthearchitecture.com

Effective software architecture sketches

3 If you're not using a formal notation (e.g. UML), make sure that other people will understand your sketches.

Titles
Short and meaningful, numbered if diagram order is important

Lines
Make line style and arrows explicit, add annotations to lines to provide additional information

Layout
Sticky notes and index cards make a great substitute for drawn boxes, especially early on

Labels
Be wary of using acronyms

Colour
Ensure that colour coding is made explicit

Orientation
Users at the top and database at the bottom? Or perhaps "upside-down"?

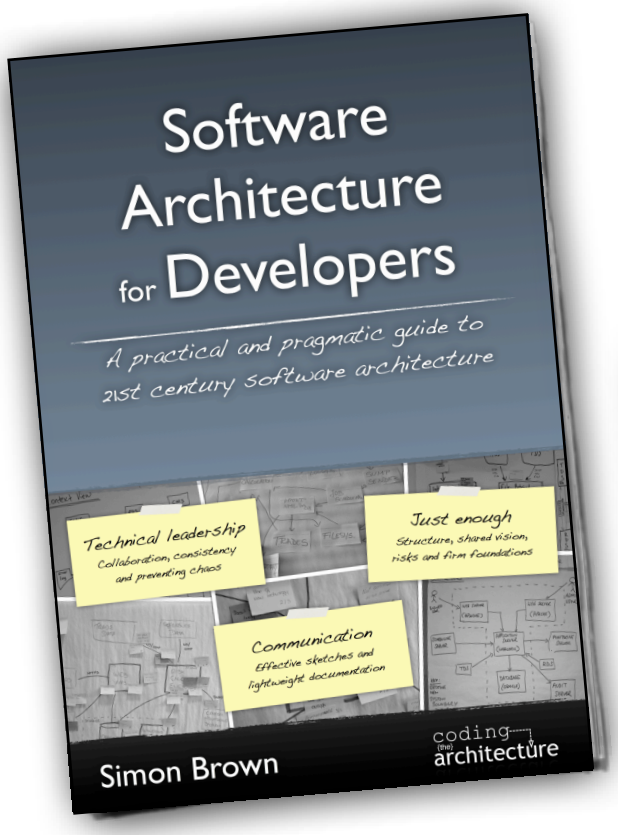
Shapes
Don't assume that people will understand what different shapes are being used for

Borders
Use borders to provide emphasis or group related items, but ensure people know why

Keys
Explain shapes, lines, colours, borders, acronyms, etc

Responsibilities
Adding responsibilities to boxes can provide a nice "at a glance" view (Miller's Law; 7±2)

Some tips for effective sketches



4 If you accidentally write on a whiteboard with a flip chart marker, just overwrite it with a whiteboard marker and it *should* erase. :-)

Published incrementally.
Variable pricing.
Buy now and get free updates from leanpub.com.

coding (the) architecture
www.codingthearchitecture.com