CRC Card Design

We use CRC card design almost exclusively. Developers, managers, and users are all comfortable with the cards, since they are non-threatening. Moving the cards around makes it easy to describe how classes work, and it is easy to throw a couple of cards away and create new ones if the design isn't working out.

We do not save the cards, nor document what they were. The classes are the documentation of the design: they represent what the system really is.

On the contrary, without a "rigorous design process", you can never create a quality system.

- Our design process is actually quite rigorous. Many developers contribute to our carding sessions, and then a team of developers realizes the design in classes. It just happens that we do not produce ancillary documentation as a by-product of our design process. We also do not waste time updating documentation, nor live with out-of-date documentation. It's a win-win approach.
- The system's quality is ensured by over 1300 unit tests. Every class has extensive unit testing that ensures it is doing what it is supposed to do.
- The system's quality is ensured by hundreds of functional tests. Each test exercises the whole system, end to end, and checks that it computes correct outputs from its inputs.

On the contrary, without a "rigorous design process", you never know what you are building or have built.

- Design documentation, in the real world, rarely reflects the reality of the software written: it is not kept up to date. Our focus is on the quality and maintainability of the actual software.
- Smalltalk's development environment lets the software stand for itself. With proper naming conventions, well-designed classes, and the powerful Smalltalk environment, it is easy to find out how anything works.

[Home] [XP] [XP Practices Frame]

ronjeffries@acm.org
http://www.armaties.com