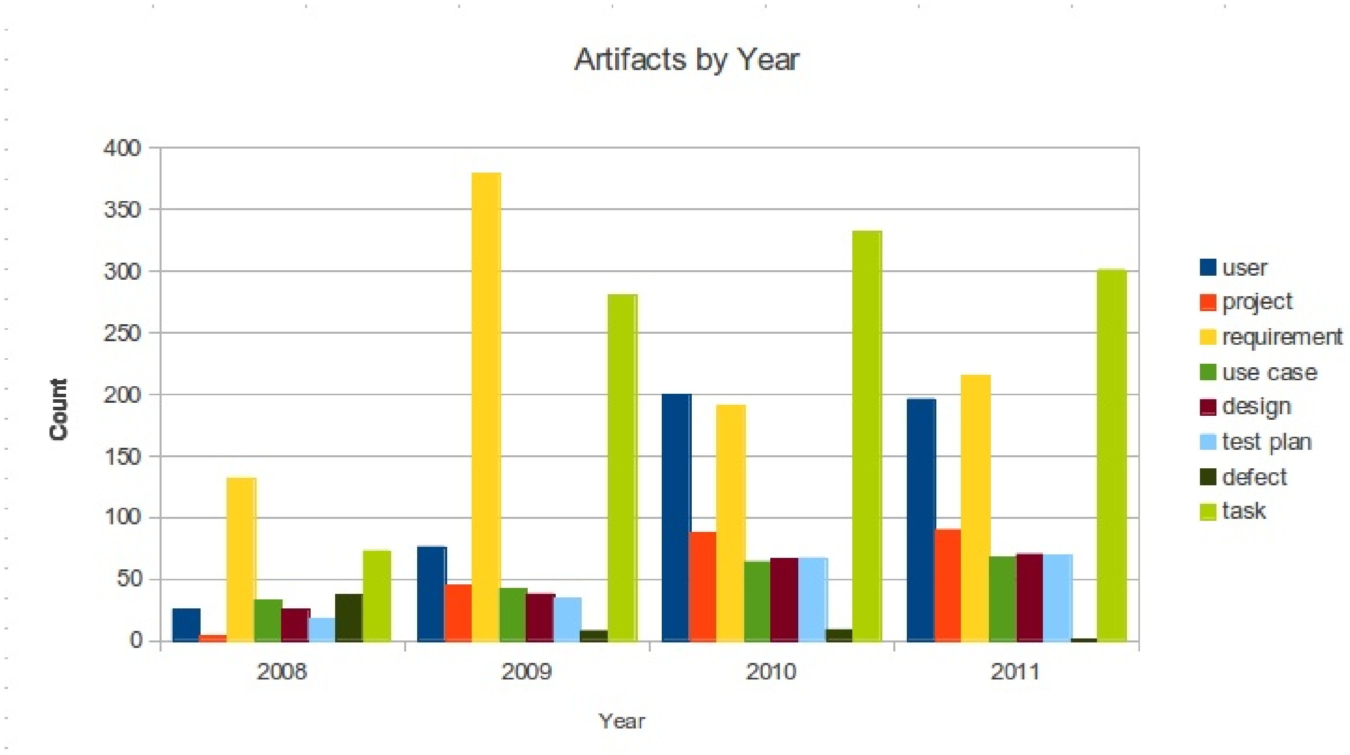
The Code Roller Project: User Activity

I started Code Roller back in 2008 because I had previously participated in the selection of the Project Management solution for the Independent Software Vendor that I was working for at the time. What I discovered was that there really was no PM tool that was focused on the Software Development Life Cycle. I believed that the world of software development could benefit from a product that was a combination of the best of Doors, Source Forge Enterprise Edition, and a light weight version of the Rational Unified Process. Throw in a team based, peer review oriented workflow and some social media and you would have a better way to code. That was the original motivation behind Code-Roller.



In Code Roller, you create projects, teams, releases, requirements, use cases, designs, test plans, and defects. Teams work on projects which are composed of releases. In each release, a collection of requirements would be implemented, Requirements would get analyzed into use cases and implemented by designs. Test plans would be used to assure quality. Defects would get filed when the implementations did not satisfy the associated requirements. Requirements, use cases, and designs would go through a peer review workflow which would schedule tasks that surfaced on the user's home page.

Requirements management was always the most used feature of Code Roller. Use cases, designs and test plans never really took off in the product. Defect tracking was a complete non-starter. Apparently, Code Roller users weren't interested in their planning solution being integrated with their ticketing system

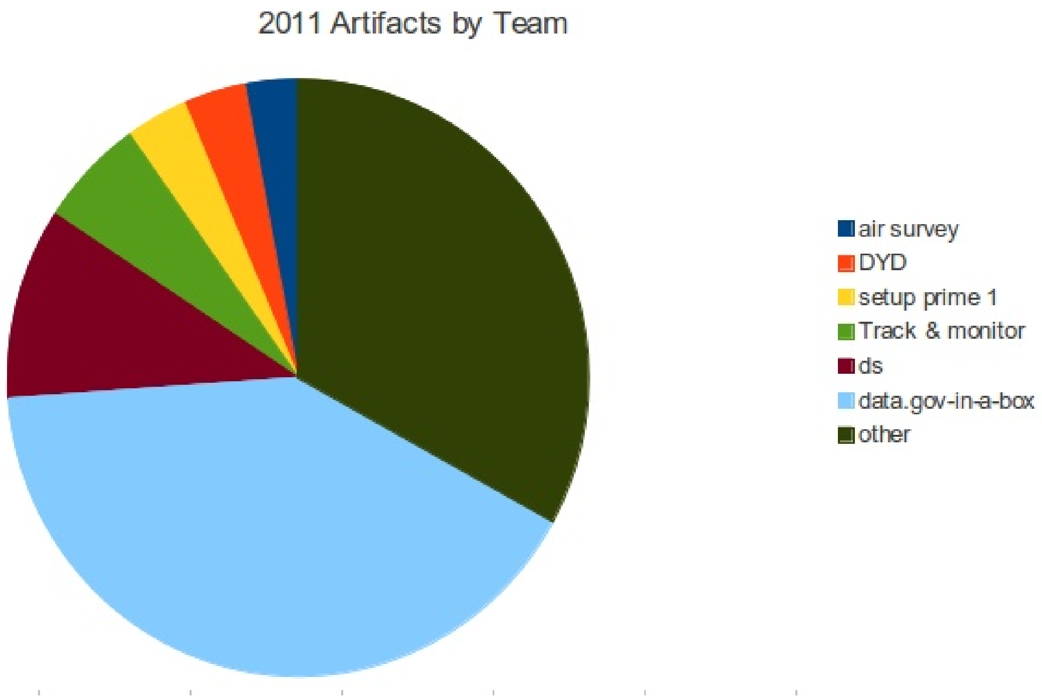
Marketing Activity

Code Roller came online in the summer of 2008. This was a side project so there was always only a minimal amount of promotion in 2008 and 2009. This promotion was in the form of discussion forums, linking from blogs, youtube videos, docstoc collateral, online tutorials, and a very modest Google adwords campaign. Special landing pages were crafted for engineers, entrepreneurs, marketing, and mobile users. No efforts were made to promote Code Roller in 2010, 2011, and 2012.

Most of these users were either students or people new to the software development process.

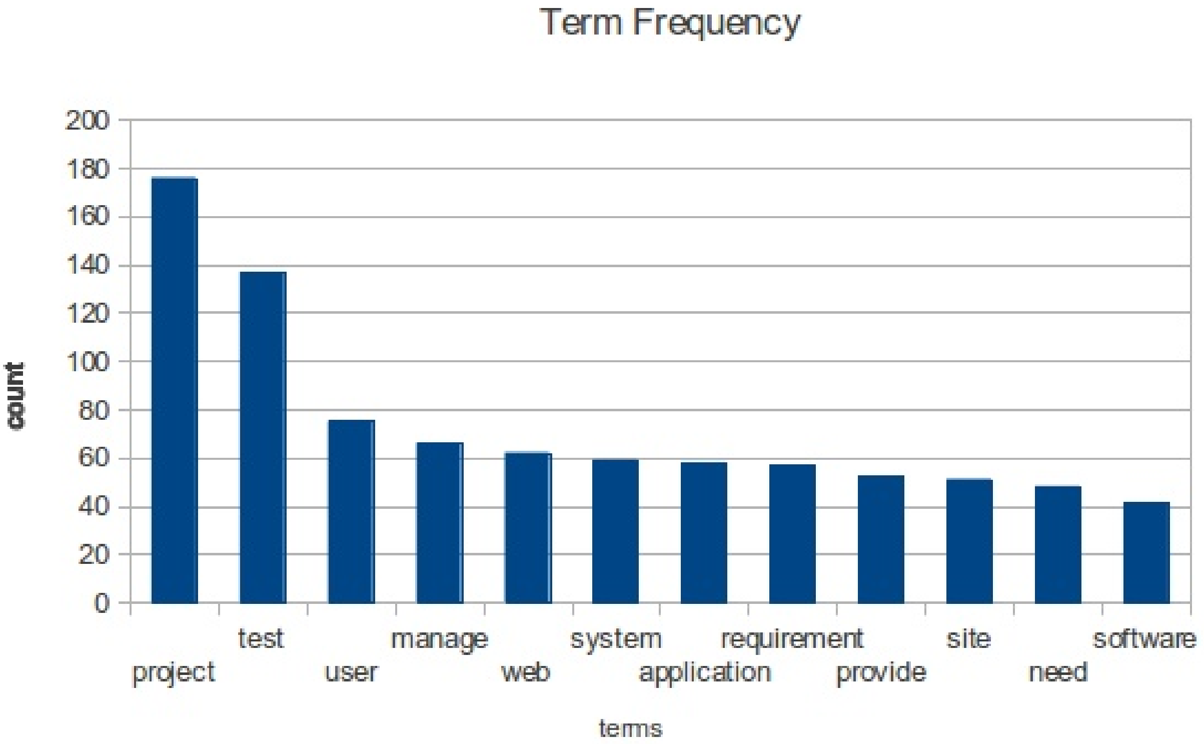
Wizard Conversion

Code Roller was all about managing software projects so the first thing to do after registering was to start a new project with the new project wizard. Slightly less than half of all registered users would do that. The wizard was a multi-step process where the user would be interviewed about the project, the team, the details to the first release of the project, and finally a starter set of requirements for that release. In 2010, abandonment of the wizard after specifying the high level project information and before inviting others to participate in the team was 4.6%. In 2011, that number dropped to 2.22%. After specifying your team mates, the user would be prompted to scope out what was to be delivered in the first release. Abandonment at this step was much higher with 14.46% in 2010 and 15.91% in 2011. Code Roller's users could not really grasp the notion that a project should be split up into multiple releases.



I suspect that most of these users were either students or people new to the software development process. In 2011, only one tenth of the active teams created 67% of the artifacts. I believe that this disparity is due to a small number of teams really using the product while most users were just exploring Code Roller with no real project in mind.

An analysis of the term frequency of the artifacts would seem to add credence to my conclusion. The most popular terms were generic in nature. The word “project” occurred the most frequently. Term frequency for the descriptions of projects, requirements, etc, followed the classic long tail distribution but clustered around the kind of generic terms that are typical in an under-graduate level syllabus for Management Information Science students.



Most coders aren't terribly concerned about a predictable software development process. What coders want to do is code. To most coders, documentation and design just get in the way of what they do best.

Code Roller's mandate was to give coders better management over requirements, last minute changes, configurations, document trail, organizational compliance, design related collateral, and releases. Most of Code Roller's users appeared to be MIS students looking to learn more about SDLC. A small number of professional teams used Code Roller as a way to capture requirements from remote clients.

What technology did I use in the making of this report? Code Roller uses MySql as its persistent data store. I loaded that data into an OLAP cube with the open source Mondrian Pentaho project. The term frequency was calculated using the faceting and text based searching capabilities of the open source Apache Solr project. The charts were all generated use Libre Office. Scribus was the desktop publishing software that put it all together.

Glenn Engstrand

@gengstrand