

Rapid Prototyping Tools

Adam Pridgen

What it is & how does it work?

- Rapid development of tools
- I call it Rapid ToolSmithing
- Process
 - Need: Decompose & understand the problem
 - Requirements: Identify small functionality set
 - Assessment: Available COTS and Frameworks
 - Process: Hybrid of Bottom up / Top Down

Case Study

- Multi-threaded, host discovery using DNS
 - Need: Fast, Furious, and Functional
 - Requirements: Fast, DNS Libs, Async. Net IO, Threads
 - Assessment: C++, Bind10, Boost Libraries
 - Process
 - Outline Functionality in Classes and Header Files
 - Begin Implementation to fill in the gaps

Toolsmithing: Development Phase

- Coding Time: Roughly 50 hours
- Debug and Testing: Roughly 30 hours
 - Bugs: 15-25 Logic Issues (Parsings, RegEx, D.S. Updating)
 - Bugs: 5-10 Misunderstood the Boost ASIO Framework
- Caveats:
 - Setting up Bind10 source to compile on Windows
 - Learning to form messages and get valid responses
 - Learning the Boost::Asio Implementation and API
 - Multi-threaded debugging
 - Optimizing the prototype after understanding APIs

Conclusions

- Creativity, Patience, Persistence, and Tenacity
- Motivation relies on small milestones
- Expectations are limited by time frame
- Tool Code quality != production CQ
- Allow more time for multi-threaded ASIO Apps
- Two Primary Classes, 1 Template, & ~1450 LOC (not counting main, Bind10, etc.)