Best Practices for Software Projects - JAD Sessions

January 2004 - Pragmatic Software Newsletters

Most of us have worked on projects that were not well organized and the requirements gathering process dragged on for months, commonly referred to as "analysis paralysis". Joint Application Development (JAD) sessions are a set of intense meetings held off-site where executives, developers, and end-users define the exact requirements needed by a software solution. By holding JAD sessions, your team can more quickly define requirements, get executive buy-in, eliminate irrelevant requirements and reduce political jockeying between team members.

Below are the keys to successful JAD Sessions:

1. **Invite the Right People** - Limit JAD sessions to 10 people or less. Of the 10 people, you must have a session leader, an executive sponsor, an end-user representative, a developer, a scribe, and specialists:

   A) The **session leader** is responsible for organizing, planning and managing the meeting (handling conflicts, political struggles, etc).

   B) The **executive sponsor** is the person responsible for approving the project financing and giving final approval to go forward.

   C) The **end-user representative** is responsible for ensuring that the solution meets the needs of their business and is responsible for approving the approach.

   D) The **developer** is the person responsible for guiding the technical solution and is responsible for prototypes. They are also responsible for guiding the technical solution as to reusability and simplification of the technical requirements.

   E) The **scribe** is responsible for documenting decisions made and clarifying and tracking action items.

   F) **Specialists** are needed from time-to-time to provide specific areas of expertise. They are not needed at all meetings, they are called upon as required.

2. **Hold Meetings Offsite** - Effective JAD sessions are held offsite. This ensures that attendees do not get pulled out of the meetings and that the entire team has committed their time to the requirements definition process. The entire JAD session process should last no longer than 10 sessions, as the system should be fully defined within that number of sessions.

3. **Create Prototypes** - JAD sessions use prototyping extensively. Creating prototypes for web-based solutions can be quickly done using Microsoft Front Page (http://www.microsoft.com/frontpage) and other tools. To define workflow and how the screens work together, products like Robo Demo (http://www.robodemo.com) can help tremendously. By creating prototypes, JAD members can quickly make decisions about the design and can quickly eliminate complex and irrelevant requirements.

4. **Deliverables** - Upon completion of the JAD sessions, your team should have a prototype of the system, a user-interface design, database schema, estimates and a budget.

5. **Final Approval** - The final step in the JAD session is to obtain approval from the executive

sponsor and the end-user representative. The final approval should be evidenced by signing the deliverables documentation.

6. **Example** - Our team just finished a JAD session for the development of a Time Sheet system for one of our clients. To view the Design Document that came out of the JAD session, go to http://www.pragmaticsw.com/newsletters/JADVIL0039.pdf. It also contains a prototype, complete with movies that show the workflow of the system. This specific JAD session lasted 3 sessions and was completed in 1 week. Once the project was approved, we used Software Planner (http://www.SoftwarePlanner.com) to manage the project.

Below are some helpful templates to aid you in developing software solutions on-time and on-budget:

- **Project Management Guidelines -** http://www.PragmaticSW.com/Pragmatic/Templates/ProjectMgtGuidelines.rtf
- **Functional Specifications -** http://www.PragmaticSW.com/Pragmatic/Templates/FunctionalSpec.rtf
- **Architectural Overview -** http://www.PragmaticSW.com/Pragmatic/Templates/ArchitectureOverview.rtf
- **Detailed Design** - http://www.PragmaticSW.com/Pragmatic/Templates/DetailedDesign.rtf
- **Strategic Planning Document** - http://www.PragmaticSW.com/Pragmatic/Templates/StrategicPlanning.rtf
- **Test Design** - http://www.PragmaticSW.com/Pragmatic/Templates/TestDesign.rtf
- **Risk Assessment** - http://www.PragmaticSW.com/Pragmatic/Templates/Risk%20Assessment.rtf
- **Weekly Status** - http://www.PragmaticSW.com/Pragmatic/Templates/WeeklyStatusRpt.rtf
- **User Acceptance Test Release Report** - http://www.PragmaticSW.com/Pragmatic/Templates/UATRelease.rtf
- **Post Mortem Report** - http://www.PragmaticSW.com/Pragmatic/Templates/PostMortem.rtf
- **All Templates** - http://www.PragmaticSW.com/Templates.htm
- **Prior Newsletters** - http://www.PragmaticSW.com/Newsletters.htm
- **Software Planner** - http://www.SoftwarePlanner.com

About the Author

Steve Miller is the President of **Pragmatic Software** (http://www.pragmaticsw.com) , the makers of **Software Planner** (http://www.softwareplanner.com) and **Defect Tracker** (http://www.defecttracker.com). With over 20 years of experience, Steve has extensive knowledge in project management, software architecture and test design. Steve publishes a monthly newsletter for companies that design and develop software. You can read other newsletters at http://www.PragmaticSW.com/Newsletters.htm. Steve's email is steve.miller@pragmaticsw.com.

Pragmatic Software Co., Inc.
9085 E. Mineral Circle
Suite 340
Englewood, CO 80112

Phone: 303.471.8355
Fax: 303.346.1749
Web site: http://www.PragmaticSoftware.com
E-mail: info@pragmaticsw.com