

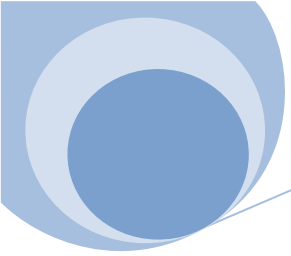


# SmartBear Software

Pragmatic Agile Development (PAD)  
Conceptual Framework

This document describes the Pragmatic Agile Development framework, a Scrum based development process.

**AutomatedQA**  
3/10/2010



## Pragmatic Agile Development (PAD) Overview

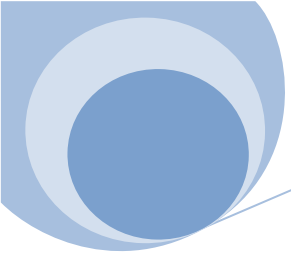
Pragmatic Agile Development (PAD) is a software development methodology based on Agile Scrum that has the following goals in mind:

- **Faster Releases** – With PAD you develop software in 30-day sprints allowing your clients to obtain a new release of your software monthly.
- **Quality Software Releases** – With PAD you develop software that is tested in QA and Beta before moving to production, resulting in higher quality releases.
- **Quicker Return on Investment** – By implementing software more quickly, you can more quickly obtain revenue from it, improving your return on investment.

## The Scrum Team

The Scrum Team consists of these roles and is normally staffed by no more than **7 resources per team**:

- **Product Owner** - This is the person that identifies and prioritizes the features that will appear in a 30 day sprint. This is normally the CEO, CTO, Product Manager or some other high level stakeholder that ultimately is responsible for shaping the roadmap of their product.
- **ScrumMaster** - The ScrumMaster is akin to the Project Manager in Waterfall environments, but does not manage the team deliverables at a micro level. Instead, this person is responsible for ensuring that the 30 day sprint stays on course, no new features are added to the sprint, that code inspections happen, and for ensuring everyone plays by the rules.
- **The Team** - With Waterfall, a team consists of analysts, designers, testers and documentation specialists. With Scrum, each team member is empowered and expected to self-manage themselves and to participate in all duties needed to deliver a feature. This includes analysis, design, coding, refactoring, testing and documentation. In Scrum, you normally do not have a Software Quality Engineer, as it assumes programmers can fill that role. In our experience, it is best to have a dedicate resource for this, as it ensures another level of protection from buggy code, so you will have a separate person that does this. So in our implementation of Scrum, the Team will be responsible for analysis, design, coding, refactoring, **unit testing** and **technical documentation**.
- **The Software Quality Engineer** - Although not in the purest version of Scrum, we see the Software Quality Engineer as a key person in the Scrum Team. Their responsibility will be to aid in developing a robust set of test cases for each work order. The programmers will run those test cases (and fix any defects) before turning the feature over to the Software Quality Engineer for QA testing. Once the Software Quality Engineer identifies that a feature has been fully designed, coded, unit tested and technical documentation has been done (if needed), the feature is marked 100% complete which triggers the Software Quality Engineer to begin his/her testing. This person will also do a weekly regression test each Friday



- **The Documentation Specialist** - Although not in the purest version of Scrum, the Documentation Specialist is the person that creates the help guides and movies for the features.

## The Product Backlog

As your existing clients request new features or as you come up with new features that make the product more marketable, these items are called the **Product Backlog**.

## PAD Planning Week

In the purest version of Scrum, you have only 1 planning day for the sprint and requirements are written on index cards (called User Stories). We believe this is not enough time or detail to deliver quality features, as we have shown that taking the time to fully detail the feature saves time once the client (or your internal team) receives the features, it takes less re-work. So you should devote a week to planning.

Each release (or sprint) will begin with a PAD Planning Week. The first day of the PAD Planning week will begin with defining the goal for the Sprint and identifying features you wish to have in the release, in priority order.

In Scrum, releases are done in 30-day sprints. The 30 days are calendar days, so with holidays and weekends, this may equate to 19 to 23 working days. The 30-day sprint begins after the PAD Planning week concludes.

Each team member will identify the number of hours they can contribute to the sprint, allowing you to determine the maximum velocity for the sprint (velocity simply means the number of hours that can be worked in the sprint). By knowing the maximum velocity, you can determine what features will fit in the sprint.

After the high level features are identified in the first day of PAD Planning, you will assign specific work order numbers to each high level feature and assign a set of work orders to each team member. The team members will spend the week defining the detailed requirements for their assigned work orders.

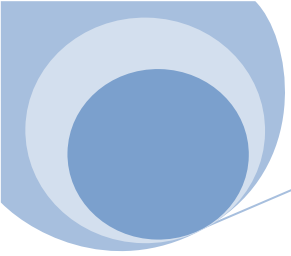
**Note:** Work Orders are simply a hard copy document of each functional specification.

### User Interface Design Guidelines

Because you will have multiple team members creating requirements, it is critical for your team to define your user interface styles in a style guide and ensure all your team members adhere to those standards.

### Decomposing Work Orders

Since sprints are limited to 19 to 23 working development days, you will be forced to make hard decisions about the features that can fit into the sprint. When you are assigned a work order for a



feature of the sprint, you should decompose the feature into multiple work orders so that you can prioritize specific pieces of the feature.

For example, let's assume that you are redesigning your user interface for a more pleasing look and feel, and the aesthetics are the most important issue for the sprint. You also would like the screens to be more user-friendly, so you have some issues you wish to address (like prompting to save changes when changes are made and they switch between tabs on the screen). In this case, you should decompose these into 2 separate work orders (one for the aesthetics and another for the tab switching). By doing this, it allows you to prioritize the tab switching lower than the aesthetics.

### 30-Day Sprint

At the conclusion of the PAD Planning Week, you should have detailed requirements and estimates and the Scrum Master will create a project plan that contains each work order and the individual assignments. The items that appear in the project plan for the sprint are referred to the **Sprint Backlog**.

Each feature will have a priority and will be worked in priority order so that if you do fall behind, you can ensure the highest priority items make it into the sprint. Features not completed at the end of the sprint can be reprioritized for possible inclusion in the next sprint.

#### Daily Code Builds

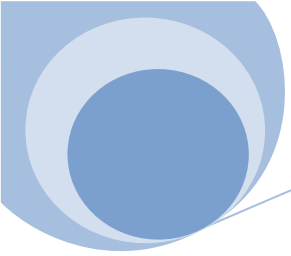
Each day, each team member should check their code into their source control system if their code is compilable (never check in code that is not compilable). If Database Script changes were made, these also need to be checked into the source control system.

It is wise to make use of an automated build tool that runs at the end of day and will do a GET on all code and SQL Script changes that were made. It will then compile the code into executables and will run the SQL script to upgrade the database. That way you will have a new build on your quality assurance server daily so that our Software Quality Engineer can test new features and run regression when needed. It is also advisable to craft a set of automated test scripts that run with each build to ensure the new build has a certain level of integrity.

#### Code Inspection

It is important to know that your code should not be considered complete until it is fully coded, fully unit tested, you have run all established test cases, you have refactored the code (if needed), and you have provided technical documentation (when needed).

Upon indication that your code is complete for a feature, the Team will do a code inspection by reviewing the code in VSS related to your feature for standards adherence, identification of logic errors or performance problems, and reusability. If the code inspection finds failures, defects will be created and assigned to you to fix the issues before the code is tested by QA.



### **Daily Hours Entry**

Each day, every team member must log the hours they worked on each task during that day. If they logged time to defects, they will update their time on the Defects project plan. When logging time, you must also enter % complete or Estimated Remaining Hours (this is the preferred method). By entering your Estimated Remaining Hours, you will know how many hours remain for all tasks and can determine if you are progressing on a pace to finish the sprint with all the desired features. Utilize burn down charts that show daily trending of Estimated Hours Remaining. A burn down chart is simply a chart that shows day-by-day the number of estimated hours, actual hours and estimated hours remaining. As the sprint progresses, you should see the estimated hours trend downwards and it should be on pace so all the committed work is accomplished in the sprint.

### **Daily Scrum Meeting**

You will hold a Daily Scrum Meeting each day. In the purest version of Scrum, this daily meeting is restricted to 15 minutes and each team member is asked 3 questions:

“What did you do yesterday?”

“What will you do today?”

“Are there any roadblocks or anything impeding your progress?”

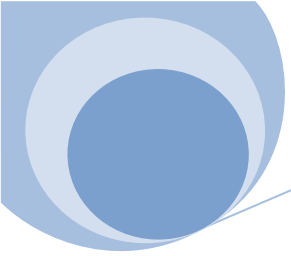
In our experience, 15 minutes is not always enough time to have a good dialog and a meaningful meeting. Many days you will complete this in 15 minutes, but most often it will probably take about 30 minutes.

In the meeting, you should use a Sprint Roadmap document that contains the release goal, statistics regarding how you are progressing and statistics regarding defects.

### **Scrum Rules**

The ScrumMaster is responsible for ensuring the following Scrum Rules are followed:

- Each team member must check their code in daily (or when it is compilable).
- Each team members must do a GET on Source Code at the beginning of each day to ensure they have the latest code.
- Each team member must enter their time daily (time worked and estimated hours remaining).
- Each team member must attend the Daily Scrum Meeting (or have a representative present).
- Each team member must arrive to the Daily Scrum Meeting on time or must pay a \$1 fine to the ScrumMaster.
- Test Cases must be created before coding begins on any feature.
- Test Cases must be run by the programmer before releasing the code for inspection.
- Code inspections must be done on all completed features.



- Once the PAD Planning Week concludes, no new requirements can enter into the 30-Day Sprint UNLESS the sprint is ahead of schedule and can absorb the new work (which is not likely).
- If during the 30-day sprint there is pressure from the Product Owner to add or change requirements because of new client obligations, the sprint must be ABORTED. When a sprint is aborted, the sprint ends and a new PAD Planning Week will ensue, followed by a new sprint.

## Beta Environment

To prevent surface level defects from making it into production, you should beta test all software before putting it into production. It is good practice to “**eat your own dog food**”. This simply means that you should use the product you are selling. If that can be done, use your own team for the Beta version so that you can personally discover any issues. It is expected to stay in beta from 2 to 4 weeks. Once the Beta has proven to be stable, move it to production so that your other clients can access it.

## Retrospectives (Post Mortems)

Each sprint will be followed by a Post Mortem, where you will document what you did well and what can be improved upon for the next sprint.

## Product Roadmap

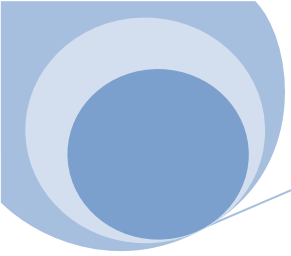
The Product Owner is responsible for creating a Product Roadmap that illustrates his/her vision for the product. The Product Roadmap shows the sprints planned for the current major release version (e.g. Release 9.0) and what major items are planned for the release. The Roadmap should be available for all team members to see.

## Professional Services

PAD is not always the best solution for Professional Services because many times Professional Services are independent custom programming that does not affect your software architecture for a specific release (this may be reports, customer installations, conversions, specialized triggers, websites, etc.). Since these work orders do not affect your architecture and releases, there is no need to bundle them into a PAD release or sprint.

Hence, consider using a Waterfall project management methodology for your Professional Services. You should still keep a separate Product Backlog for Professional Service items that are **not yet approved**.

Once a Professional Service work order is approved, it should be moved to your work list, worked in priority order, and will be moved to production upon successful QA and/or User Acceptance Testing.



**SmartBear Software**  
<http://www.PragmaticSW.com>  
<http://www.SoftwarePlanner.com>



## **Learning More about PAD**

If you are interested in learning more about PAD and how it can help your organization, AutomatedQA offers on-site training for PAD.

The 2 day course covers the methodology, templates to get you started, practical ways to use Software Planner (or other ALM tool) to track your software releases, and best practices for implementing the methodology. The cost of the on-site course is \$4,000 US plus travel and per diem. The cost includes up to 10 participants.

**To learn more about PAD or to sign up for training, contact AutomatedQA at 800-773-7785 or +1-303-768-7480.**