

# Code Sprinters basic info pack

Prepared by: Andy Brandt

April 17, 2008

# Our approach

## Agile development

We are a fully agile company and we use agile methods in both project management (Scrum) and our software engineering practices (Extreme Programming - XP).

Agile approach focuses on providing working, tested code of high quality from the very beginning of the project assuming the scope to be changeable over time. This way the project's product is not assumed to be fully envisioned in every detail at the outset, but rather evolved and enhanced over time as the client's knowledge of the product increases.

Thanks to that approach agile teams excel in delivering high quality software products in environment rife with change such as startup companies. High emphasis on testing and client involvement in the project results in high-quality software produced much faster than with traditional, waterfall methods.

See [resources](#) on our web page, including a [my presentation](#) about how it is to work with us to get more info about these concepts - or talk to us.

## Sprints

In all engagement modes we work in sprints – short, time-boxed iterations of work.

At the beginning of each sprint the list of all the things planned for the system being developed (called "product backlog") is reviewed with the client and sorted according to current client's priorities. Then a set of items of total size depending on team's capacity is selected from the top of the list (highest priority) to be developed within coming sprint. Then the team sets off to work to deliver this set of items while client is free to alter and edit the product backlog before the next sprint will begin.

"Time-boxed" means that sprint's duration cannot change – the deadline is unmovable. The quality level is also fixed – whatever is done at the end of each sprint must be of production quality and conforming to our definition of done. This definition is: implemented, covered with unit tests, tested functionally, documented where applicable, ready for deployment to production server.

Essentially the product of each sprint is therefore a new version of the software being developed, potentially ready for public deployment. Whether it is actually deployed is up to the client to decide.

Sprints have always the same length in a project, between 1 and 4 weeks. In our case it is usually 2 weeks and experience shows that it is indeed the best duration for web applications development. This relatively short sprint duration is possible because of the fact that web applications have little release overhead as compared to desktop (compiled) systems.

To be precise, we use work days for counting the sprint duration. In allows us to maintain the work rhythm since all sprints last the same number of working days.

Because the sprint duration is fixed a sprint calendar extending into the future is set up at the outset of the project allowing for long-term release planning knowing in advance when sprints end and thus new software version is available.

## Tools for the clients included

As part of our engagement we provide each of our clients with a following set of tools to allow them to actively monitor and steer our work on their project. Client gets access to all of them upon project start:

- **Project code repository** – this repository, kept using Subversion, is the place where developers deposit new version of the code they work on many times a day, here also all releases are tagged etc. The client can view or download the repository at any moment.

The repository is accessible using a web browser from the project Wiki pages or using one of many free, open-source Subversion clients.

- **Project bug tracking & Wiki pages** – we track bugs and other issues as tickets in a system called Trac. This system also provides project Wiki – a place where information relevant to the project is kept including contact data for the team and the client, links to test system and relevant internet pages, long term ideas and documentation to more complex parts of the code.

The issue tracking system and the Wiki pages are accessible using a web browser. All pages in the project Wiki are accessible and editable by the client.

- **The test system** – it is the working version of the system under development with test data running on our development server. It is the most current version of the system under development automatically updated from the repository many times a day.

The test system uses test data to provide our team and the client with the opportunity to follow changes in the functionality of the system being built. It allows all interested parties to follow the evolution of the system day after day, detect problems and provide input.

- **Scrum tracking software** – it is the place where product backlog (list of all functionality to be done) and sprint backlog (list of things under development in a given sprint) are kept, where the developers update daily task estimates and the progress is monitored. As a client you will get account in this software application that will enable you to manage the product backlog and follow the developers' progress in the current sprint as well as review the history of completed sprints.

For Scrum tracing we use our own Ruby on Rails application: [Banana Scrum](#). It also available as a hosted tool, so you can register and use it to run your other projects.

At all time during the project you'll have access to all the resources mentioned above. You'll be able to track the status of each task in each sprint, as well as the overall progress. You'll be able to play with the test system at all times - it is the system we develop that is updated many times a day during development and therefore all the features developed can be seen there. Last but not least, you will be able to review the code in the repository should you want to.

On top of the above you'll have direct access to the developers working on your project through Skype, phone and e-mail. You'll be encouraged to contact them with any questions and ideas. We'll ask you to follow the development in the test system as frequently as possible, also our developers will contact you with any questions and clarifications they might have with each backlog item (functionality) being developed.

## Project initiation

Before we start a project we do the "initial backlog creation" to capture the first list of things to be developed. It is a 1-4hrs session during which we will assist you in breaking down the requirements or vision that you have into "user stories". User stories are a plain English description of what a given type of user (a role) should be able to do with the system and why (eg. "a user should be able to log in using username and password").

A list of those is created which is sorted by you and estimated by us. This allows us together to see how your project will go, asses its size and do longer term planning - estimate when the site or system could be launched and then what features will come when with the upgrades it will get after each sprint. We also discuss technology choices, team size and how the releases will be done in terms of responsibility, etc.

With larger projects only the most important stories might be broken down in detail during this first session, less important might be left as bigger-sized items to be broken down later. The idea here is to provide enough information just in time - stories that are to be implemented within next few sprints should be described in enough detail to allow design & coding based on them. Stories that will be addressed later can be left as more general descriptions.

When we have the agreement signed we set the length of sprints (iterations) and their dates. Then we start with a 5-10 work days "Sprint 0" that is devoted mainly to project setup. It covers:

- set up of a repository for the code,
- set up of a test system, including the design of the basic initial data structures,
- set up of a test system for the legacy code/system if there is any (none when a new project),
- set up of the project's bug tracking system (Trac) and Wiki,
- set up of the Scrum sprints assistance system as discussed above,
- discussion of and setting up the deployment procedure (how the result will be deployed to production),
- initial acquaintance with the project team + set up of communication channels (Skype, phones, e-mails),
- final prioritization of the initial backlog before the first sprint,
- initial development of the foundations for the new system.

At the end of "Sprint 0" we expect to have a ready development infrastructure plus some basic development done. Then, the development will start according to normal rules & procedures of Scrum. As discussed above sprints usually last 10 work days, the last day of each sprint is the release day: completed functionalities from given sprint will be available, potentially for deployment to the production system.

In most cases the Sprint "0" does not last longer than the typical 5 days.

# Commercial

## Payment models

We offer two basic payment models: **per hour** and **per sprint**.

In the **per hour** model each developer working on your project registers his time in a special app describing day by day what he was doing and how long. At the end of each month you are billed for all the hours worked by all the developers over that month according to the hours registered.

In the **per sprint** model you pay one flat rate for each developer assigned to your project for each sprint. Hours are not registered (except for the purpose of tracking progress in the Scrum tracking app, but that's a bit different).

Each developer assigned is assigned exclusively to your project for each sprint in both models. In both models you can stop the project at any moment. In that case in the per sprint model you have to pay for all the sprints started, in the per hour model for all the hours that have been worked until the cancellation came in the given month.

Changes to the number of developers assigned are, of course, discussed and agreed with you. However, if you don't say anything we assume we can continue the next sprint with the same team size as the previous one.

Of course, everything we develop is legally owned by you.

Attached is our simple agreement for the "per sprint" model, its "per hour" version differs just slightly. Please note that the legal name of the company is "ITTC Andrzej Brandt" as it is privately owned at present. We plan to establish "Code Sprinters LLC", but it has not happened yet.

## Rates

Below are our standard rates. VAT tax has to be added where applicable, especially for clients registered in Poland.

| Team Member      | Per Hour | Per Sprint |
|------------------|----------|------------|
| Developer        | €38      | €2,280     |
| Senior Developer | €49      | €2,900     |
| Junior Developer | €31      | €1,800     |
| Tester           | €24      | €1,440     |

Project management is included with our own team. However, if we are to manage external resources (for example your own developers) project management is added as an explicit additional service at €55 / hour.

This agreement is made and entered into on this \_\_\_\_th day of \_\_\_\_\_, 2008, between ITTC Andrzej Brandt (EU VAT number: PL5211309613), located at Swietochowskiego 3/58, 01-318 Warszawa, Poland (hereinafter "CONTRACTOR") and \_\_\_\_\_ (hereinafter "COMPANY"). Whereas COMPANY and CONTRACTOR hereby enter into a contract and agreement whereby CONTRACTOR will render certain work, services, labor, and/or materials to and for the benefit of COMPANY for valuable consideration.

NOW, THEREFORE, for and in consideration of the mutual covenants contained herein and other good and valuable consideration, the receipt and sufficiency of which is hereby acknowledged, COMPANY and CONTRACTOR do hereby contract, covenant, and agree as follows:

1. CONTRACTOR hereby agrees to provide to COMPANY the following services: software design, development, testing, and technical project management.
2. Services will be delivered in iterations of set duration of 10 working days (public holidays in the Republic of Poland as well as Saturdays and Sundays being considered non-working days) called SPRINTS.
3. COMPANY hereby agrees to compensate CONTRACTOR for the services rendered at the following rates per SPRINT per each CONTRACTOR developer assigned to the project for services rendered:  
\_\_\_\_\_/ programmer / sprint + applicable VAT tax,  
\_\_\_\_\_/ tester / sprint + applicable VAT tax,  
\_\_\_\_\_/ PM/technical lead / hour + applicable VAT tax.
4. The number of developers assigned to deliver services to COMPANY in a given SPRINT will be always agreed between the parties before the beginning of each SPRINT.
5. Such payment will occur within fifteen (15) days ("net 15 payment") of receipt of CONTRACTOR'S invoice for completion of said services by CONTRACTOR.
6. CONTRACTOR will invoice COMPANY monthly for all the SPRINTS completed in a given month.
7. Rates may be renegotiated at any time subject to the agreement of both parties, such new rates to be attached as an Exhibit to this Contract.
8. CONTRACTOR shall provide and use its own equipment for purposes of performing the duties of this Agreement.
9. CONTRACTOR agrees it is not an employee of COMPANY for any purpose whatsoever.
10. CONTRACTOR agrees that it shall complete the work services or labor required under this Agreement in a workmanlike manner and shall further keep all property of COMPANY free and clear of all liens and encumbrances.
11. CONTRACTOR will take reasonable precaution to keep confidential COMPANY's trade secrets and other sensitive information that was made available to CONTRACTOR while executing this contract.
12. No waiver of any provision of this Agreement shall be deemed, or shall constitute, a waiver of any other provision, whether or not similar, nor shall any waiver constitute a continuing waiver. No waiver shall be binding unless executed in writing by the party making the waiver.
13. All intellectual property created under this contract remains the sole property of COMPANY. CONTRACTOR may not sell, distribute, or reuse any code created during the performance of work performed under this contract without the written permission of COMPANY.
14. In the event that payment for services is not rendered in full, the intellectual property created under this agreement will revert to CONTRACTOR when any payment becomes 180 days overdue.
15. Without prejudice to any other right or remedy, CONTRACTOR may charge interest on any sums due from the COMPANY to CONTRACTOR which are not paid by the due date at the rate of 4% above the UK Barclays Bank Plc base rate accruing daily, before and after judgement, from the due date for payment until the payment has been made.
16. This contract encompasses the entire agreement of the parties and there are no other agreements, oral or written.
17. This contract may not be modified or amended except in writing with the same degree of

formality with which this contract has been executed.

18. Should any part of this contract be adjudicated inoperative or invalid, the remaining provisions of this contract will remain in effect and operate as if the invalid or inoperative provision had never existed.
19. The construction and interpretation of this contract and all transactions under it shall be governed by the laws of the Republic of Poland.

WITNESS our signatures this \_\_th day of \_\_\_\_\_ 2008.

BY: \_\_\_\_\_ AND \_\_\_\_\_