



Evaluating Software Development is Measuring the Invisible

September 19, 2013

By Dmitry Yakovlev

Recently I read a blog post by McKinsey titled "[Enhancing the efficiency and effectiveness of application development](#)." It focused on building yet another universal framework for measuring efficiency of the software development process.

This is unfortunately a popular theme for many from the community who often rely on ideal world scenarios of non-changing requirements and non-sliding deadlines.

Most of the project managers will admit that running IT projects without any metrics is evil, and I completely agree with them. However, I should note that all widely-adopted software development methodologies do have such metrics to ensure certain sanity level at least.

What surprises me is when the value an engineering team brings to business is measured in lines of code or story points. Having been working for a vendor providing custom software development services for a long time, I strongly disagree with associating the value a software team contributes to business with X use case points delivered over N months.

Why do organizations undertake application development projects? Because they feel pain or see new opportunities. Solving these problems defines the success of the project. The result is always a joint effort of the engineering team and business stakeholders. From such an angle, the most valuable skill of a development team is ability to collaborate with business.

With the understanding that the role of software developers is not just earning story points from coding, let's take a closer look at the values a good engineering team brings to the table.

- **Facilitating communications with stakeholders**

No matter the functional specifications, there are many details that can be discovered only in the process of daily communications between developers and business stakeholders. Until these details are discovered and implemented, the project won't be perceived by users as successful.

- **Challenging business requirements**

Development teams help business people to better understand what they want by challenging their wishes, thus do a good job by refining and prioritizing requirements

- **Creative thinking and proactively**

Developers often come up with bright ideas about improving application functionality and design as well as optimizing business processes within the organization basing on past experience, specific knowledge or good understanding of problematic area.

In addition to what I listed above, engineering teams capable of creating sound emotional atmosphere around the project greatly facilitates the whole process and bring extra value. Challenge, inspiration and motivation — all of these intangible assets can't be measured in story points, but if combined they define overall success.

With that said, the next logical step would be proposing a valuation model accounting for these intangible things. I do not consider high-level definition (which would be loved by executives) of a value brought by IT project as a change in company's net profit. Apparently, it's hardly possible to calculate such a change.

I also doubt we can rely on use case points to compare different projects even inside the same company. First, understanding of what is one point varies from project to project. Second, use cases are applicable mostly to

user-interface. Enterprise applications can't be described via use cases only because there is a lot under waterline. Imaging two functionally identical simple web applications (for example, displaying current date and time), the first one serving just couple of users and second one serving millions. The amount of effort required to deliver these two projects will be totally different, while a number of use case points earned will be the same. DataArt, as a custom software service provider, has dozens of engineering teams running in parallel. There were a number of attempts made on elaborate metrics to measure how healthy the teams and clients. No surprise, all story point-based approaches failed miserably, being not able to identify problematic projects. Finally, we arrived to a set of simple questions business stakeholders are asked when we need to assess how they value the outcome of the project:

- Did project go live at right time?
- Did you organization notice the impact from the project?
- Do end users use the system? Do they like it?
- How many issues were filed by end users after going live?
- Do you consider releasing a next version of the application?
- Do you think the team did all their best? Did you congratulate the team?
- How often did you talk to the team? Do you know all engineers by name?
- How many times did you change your mind after talking to the team?
- Would you undertake another project with the same development team?
- Would you recommend this team? Did you recommend?

This list is not complete but it gives an idea what to look at when measuring value of application development projects. I hope it's not too late to ask such questions in your projects.

Read more at <http://venturebeat.com/2013/09/19/815276/#UhIY2R75IBMfPu6S.99>