The user story considered harmful

James Robertson & Suzanne Robertson – *http://www.volere.org/*

Agile techniques have brought us many advantages and good ideas – unfortunately, the user story is not one of them.

The user story is a “placeholder for a conversation”, or a “placeholder for requirements”, either definition is acceptable. However, if the story is a placeholder, then it must hold the right place, and must guide the conversation in the right direction. Many stories don’t.

What’s wrong with the user story?

The most fundamental problem facing software development today is that *the single greatest cause of project failure is a failure of requirements*. This “failure of requirements” covers the full gamut: failure to discover the needed functionality; failure to understand the nuances of the needed usability; failure to adequately convey the requirements to the developers; and frequently, failure to uncover the real problem to be solved. Sadly, the user story often directly contributes to the requirements problem.

So why are user stories poor in the requirements arena?

Let’s start with the whole idea of the *user*. When someone writes a user story, they have already *presumed* what the solution should be, *presumed* who will be using it, and *presumed* the functionality of the solution. In other words, the user story writer thinks he knows the solution, but probably doesn't understand the problem this is meant to solve, if indeed he knows what the real problem is.

If we look at the staggering number of projects that deliver incorrect solutions (there are numerous studies that confirm this), we can safely conclude that these projects failed to correctly understand the problem, and yet pressed ahead with a solution.

Anatomy of a story

This is the user story in its most common form:

> As a [role]
> I want [functionality]
> So that [business value]

A story like this does not always, and more often not, indicate what business problem it is trying to solve.

The reason is this: "I want" is almost always followed by a presumed solution: "I want to access my account from my mobile"; "I want to drag and drop scheduled items"; "I want a screen to show me train times"; and so on. These are solutions, and give little indication whether or not they solve the real business problem.

The last line of the story, the justification, "So that [business value]" might give a better indication of what the problem is. However, by writing a solution, the business value is likely to simply (and erroneously) justify the solution. The justification might appear to be valuable, but rarely is:

> As a bank customer
> I want online access to my account
> So that I can see my balance 24/7

The "So that" is justifying the online solution, but being able to see your bank balance 24/7 does not solve any real business problem, either for the customer or the bank. Seeing your balance today does not tell you how much you can afford to spend before the next payday deposit. Seeing your *discretionary* balance is more helpful, because without knowing your commitments for the rest of the month, the current balance is of little use.

When there is a user there must be a solution for the user to use. The solution is not the place to start, particularly when it is merely a presumption.

The often-quoted INVEST (Independent, Negotiable, Valuable, Estimable, Small, Testable) criteria for stories are useful when it comes to planning sprints, but apart from "Valuable", they do little to ensure that the story is in fact solving the right problem. Showing the value of a story is of course, er, valuable, but any story can show a value and still be the wrong story. The value must be value to the business, and not a justification of the chosen solution.

The only useful software we can develop is that which contributes to the business that deploys it. So instead of writing about users, we suggest that you start your stories by writing about the business. So let's call this a *business story*, and using the same format, you can write:

> As an [external customer or other external entity]
> I can [achieve a business goal]
> So that [value to the external customer / entity / business]

Note that the business story does not attempt to provide a solution to the problem, but talks about a business goal that provides value to an external customer, and thereby to the providing business.

To illustrate this, let's revisit the banking example, and this time we look at the problem from the business point of view and omit any possible solution to the problem. It gives us a story like this:

> As a bank customer
> I can have frequent and convenient connection with my account and
> its activity
> So that I can feel confident that I always know my financial position.

This story we suggest, delivers real value; if the bank enables customers to always feel that they are in control of their banking situation, then the bank is likely to attract new customers, and make existing customers happy. There is more to come from this story: it does not give any detail on how the "frequent and convenient connection" is to be achieved – but this is to the good. We now have a story which is not only holding the right place for a conversation, but has changed the conversation to be a design conversation. "Now that we understand the problem, what is the best solution?" In a few moments you can come up with a list of possibilities such as:

- A text alert whenever there is any activity on the account

- Projected balance when the direct debits and standing orders have been taken into account
- Alerts when the account risks becoming overdrawn
- An app that updates itself in the background so the account holder can open the app at any time and see the account activity
- Simple balance and projected balance over the telephone
- Scheduled text messages to give account activity
- Automated emails ate the end of each day/week

and other possibilities. Note how some of the above would provide real value, and would probably not come to light if we stuck with the original presumed solution of "online access". Some or all of these would contribute to the customer "feeling confident", and provide far more value to the deploying organisation (the bank) than mere "online access". This business story is holding the right place for the conversation.

Good stories talk about value to the business, not solutions.

If you are to provide real value, then that value must be aimed outwards. Consider this example that looks at the business from the outside:

> As a reader of technical articles and blogs
> I can establish a subscription
> So that I can be provided with a customised and restricted digest of links to material that I will find interesting.

The story, as we have said, is a "placeholder for a conversation". But instead of having a conversation about some automated solution – the screens, the buttons etc. – now we can have a conversation about the business problem. In this case a reader wants to have a subscription so that she gets pointers to material on the web and is most relevant to her interests. Moreover, she wants to put a limit on the amount of material she receives. Sure the reader could use a search engine to find material, but the proposition here is that the subscription service would have manual curators, and be able to provide the top five (or whatever number she wants) articles, which is not something the reader could do for herself without firstly reading all the articles.

This story is about starting a subscription; so let's have a conversation about subscriptions.

We could go down the conventional route and have the customer go online and setup a subscription, name, address, password, and all the other usual stuff. Or, because we have not specified a solution, we can talk about alternative ways of having a subscription. For example,

- Could the business attach links to technical articles along the lines of, "If you liked this article/blog, click here to receive links to similar information"?
- Could the company buy readers' technical search data from Google/Bing/Yahoo/DuckDuckGo etc. and mail potential subscribers with a starter list and an invitation to subscribe and refine the list?
- Could blog writers be paid a bounty to direct their readers to the subscription service?
- Could mobile users be given a customised service of mobile-friendly material only?

There are lots of solutions to business problems, but you are not going to find them if you leap into the first one that comes to mind, or passively accept the first solution proposed by the stakeholders.

With this in mind, you can see that the story would have been even more useful if it had been written in an even more technologically agnostic style:

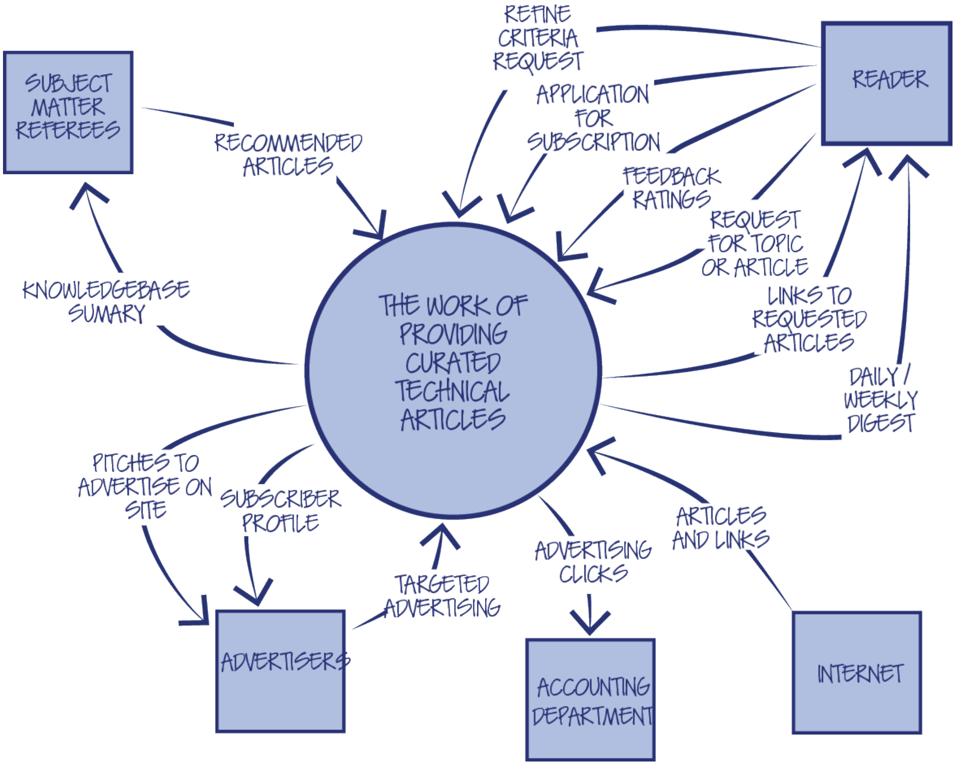
> As a reader of technical *material*
> I can be connected to a knowledge base
> So that I can receive information which is most relevant to my
> technical interests

Obviously you need to find the best solution, but you can only do that when you have a correct understanding the real business need. And by doing so, you provide real and lasting value to the deploying organisation.

A business story is not intended to be implemented; it usually contains more functionality that you could fit into one sprint. Also, it is not describing an implementable solution – it is describing the business problem and the desired outcome of solving that problem. The implementable stories come as a result of the conversation, which determines the most attractive solution. These solution stories are smaller in terms of functionality, and also involve the user.

Observing the Target Environment

Before we go any further, we should ask how we *know* that we need functionality to establish a subscription – we can't just guess. Step back a little and you can see the overall business:



Note that in this diagram you are not looking at a solution, but all of the business activity concerned with providing a technical subscription service.

If you build a context model such as the one shown above, (or some other model that shows the inputs and outputs from and to the outside world) you would see that each input or output represents a fundamental business activity: *Customer establishes a subscription*; *customer refines selection criteria*; *subject matter referee recommends articles*; and so on.

The flows of data to and from the outside world are connected to *business events*, the fundamental driver of all business activity. You can write a *business story* for each of them. The business story must provide the value for both the external customer and the owner, so now the conversation about the story can focus on the best business solution to achieve that value.

There are of course other ways of describing the response to each of the business events. If you wish to use stories, then we are suggesting you write the story about the business to be done, and avoid anything to do with a solution. The automated solution will become apparent once you have correctly indicated the real business needs.

> *"Much of software engineering is about building systems right; requirements are about building the right system."* – Bertrand Meyer, *Agile! The Good, The Hyped, and the Ugly*. Springer, 2014.

Initially, this might take slightly longer than leaping into a presumed solution. But keep in mind that according to Gartner and numerous other studies, most of the presumed solutions are the wrong ones. But by taking the little time needed to understand the real problem, the automated solution you deliver will in fact bring about real business value, and not have to be reworked before it becomes acceptable.

Writing Better Stories

So how can you write better stories? Start by not calling or thinking about them as "user stories". Put aside the user for the moment, and think about the bigger picture, the external customers and the owners of the deploying organisation. Think about the business, and the needs of the business.

Don't write "I want". Write "I need to be able to", or simply "I can". Either are naturally followed by some functionality or business activity — not by a solution. You might also consider temporarily leaving this part out of your story and concentrating on the value part.

And the value must of course be real value — both to the external customer and the owner of the deploying organisation. It must be an actual, measurable benefit. Naturally enough, the value statement connects to and supports the objectives of your project.

Don't worry about writing stories to fit into sprints. Those come later once you have written a business story that reflects the true business problem. The later stories are then about the solution, and it is the solution to the right problem.

The next time you find yourself writing a story aimed at a user, step back, and see the story from the point of view of the business. By writing business stories, you will make your eventual users much happier.

Your Authors

James & Suzanne Robertson have helped hundreds of companies improve their requirements techniques and make their system development far more effective and relevant. Their courses and seminars on requirements, business analysis and design are widely praised for their innovative approach. The Robertsons are principals of the Atlantic Systems Guild, a well-known consultancy specializing in the human dimensions of complex system building. They are co-authors of *Mastering the Requirements Process – edition 3, Getting Requirements Right*, (Addison-Wesley, 2012) and the developers of the Volere requirements techniques. They are currently working on *Minimum Viable Knowledge*, a project to demonstrate the minimal requirements, architecture and project management information that projects can develop, yet still produce a successful result.

james@systemsguild.net suzanne@systemsguild.net www.volere.org