

# -Hello decorator, how long does it take to paint my flat? - Let’s me think, about 15 strawberries.

Agile introduced estimating work in the story points. Story points estimate complexity of the project, not the time needed to finish task. In theory it makes sense. We don’t know which team member will be doing the task so estimating in hours can be not accurate. Senior developer will resolve item in couple hours, junior maybe will need more time. So, we are estimating complexity of the tasks and later we will decide who will be working on it.

**Theory is great, and what is the practice?**

I was in two teams where we were trying to implement story points. Once we had two major polish agile consultants to help us. Both times unfortunately we decided to go back to man-days. What are our learnings?

**Business needs**

When I am asking for the car repair after accident, I am not interested how much complicated the process will be, I just want to know when and how much it will cost. In IT situation is the same, if business ask for new feature he is not interested if it will be complex or not, he wants to know how much he will need to wait for it and how much it will cost him. Story points are abstract currency which tells him nothing.

**Roadmap planning**  
In every company IT together with business plan major features which should be delivered in the next year. So even if team estimates in story points one day, they or their team leader will be asked how much time they need to finish.

**People resistance**

Story points is abstract currency. The idea is that you start with some exchange rate (for example 1SP is simple control) and during the time team will synchronize and teach each other what means 1SP, 2SP and so on. The process is difficult, and it doesn’t have quick feedback loop. Often you need to spend couple months to setup some rules and understanding. And let's be honest, usually our teams consist of different kind of people. We have highly motivated guys, which will be focused on task for half year, and we will have detractors which will vote for resigning from new estimation methodology.

**Inflation & Deflation**

To introduce story points, you are using them to estimate first functionality for the sprint. After iteration finishes you check how many story points you delivered. This is the base for the next sprints. During next planning you take into sprint no more than previously finished. Seems simple. Unfortunately, we are facing deflation and inflation of the story points. During couple months one story point starts to mean completely different value. At the beginning in the sprint you could deliver 20 story points but next sprint it could be 12 or 90 with the similar job done.

**Task assignment**

Story points shows the complexity. The idea is that senior developer will finish task faster than the junior one. But I don’t agree with that thesis. Most of the time work like adding the screen, calling endpoint returning data is done in similar time by all developers. With more complex tasks I can agree that senior could be more efficient than junior, but for these tasks usually we are doing assignment during the estimation as this will be the most difficult item in the backlog.

**Hidden productivity stealing’s**

Story points also hide very important information about cost of work. If you say that something is 4 SP it is just a number you cannot say that this is long or short, but if developer tells you that changing the label on the screen costs 8 hours, you could start asking questions. This often leads to hidden difficulties which you can fight against to improve the productivity.

**How much the work costs?**

Mature developer should understand cost of tasks. I am expecting senior developer to answer question about time needed. It doesn’t need to be very precise but using story points won’t let developers improve in this area.

Agile estimation is not easy, but trying it learns you a lot.