Function Point Sampling Holds Promise for Software Metrics

As we embark on a new year 2017 which is also the 30th anniversary of IFPUG Bylaws, the software development industry is making progress. The 2015 Standish Group CHAOS report shows that in a survey of over 10K software development projects, those using agile techniques are, on average, 3x more likely to be successful than waterfall projects. The not-so-good news, however, is that the overall number of successful projects (defined as on-time, on-budget and with a satisfactory result) still remains over the past 21 years of CHAOS Reports at just under 40% of projects. The top 3 success factors in the 2015 report were not technical: 1. Executive Support, 2. Emotional Maturity and 3. User Involvement (Agile processes ranked #7.)

“On-time, on-budget and with a satisfactory result” is directly tied to the effectiveness of the estimation process. To increase the number of projects that are “on-time, onbudget and with a satisfactory result,” we need to get better at estimating and managing customer expectations about those estimates.

As the sidebar, Software Sizing Infographic by QSM, illustrates, an ignorance of software size leads to bad estimates. All estimation approaches, whether role-based, task-based or scope-based (i.e. using a parametric tool like SLIM® or COCOMO®), require an explicit or implicit understanding of what you are going to build in order to be effective. Supporters of the ISO standardized IFPUG function points (FP) know that this is where our measures can play a major role, but this unfortunately is not (yet) a universally accepted fact in the IT industry.

The Challenges with Function Points

Although the IFPUG counting rules have stabilized in the past 10 years and become an ISO standard (a good thing!), function points today face an uphill battle. The biggest users of FP do so contractually (Brazil and Italy boast the largest number of IFPUG members due to governmental regulations requiring FP.) Companies in the U.S., when introduced to function points, either have never heard of it (common) or reject the notion of using them outright based on past negative experiences with the measure.

Past negative experiences or a reluctance to try FP for the first time is often due to the fact that IFPUG FPA can be time consuming and labor intensive and requires highly specialized knowledge of FPA. Other popular sizing methods, such as counting agile story points or source lines of code, require less time and effort, but lack any agreed upon standard. Because each project team can have its own definition of story points, it is difficult to do any meaningful comparison between projects or to leverage historical data.

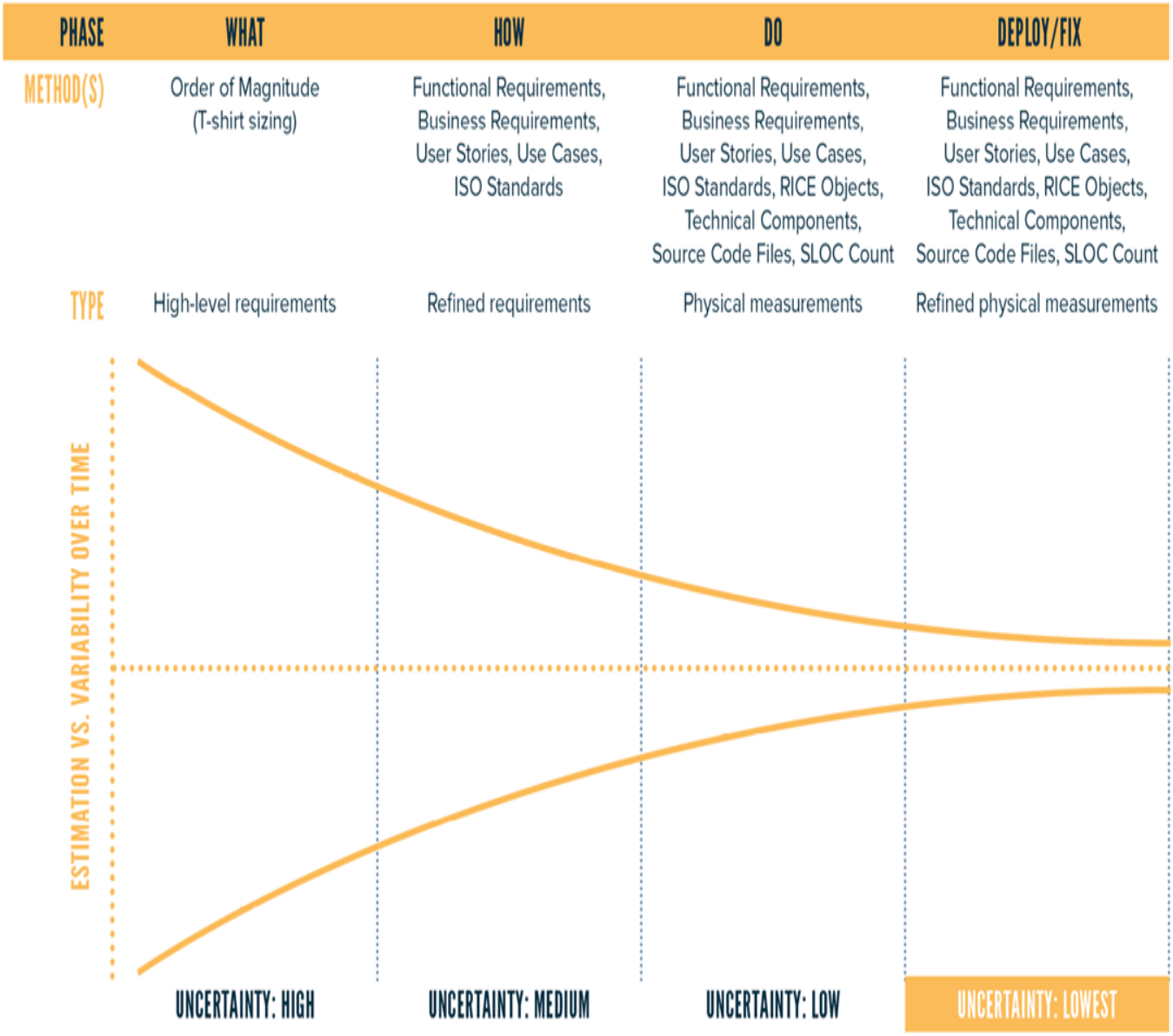
So how can we leverage the robustness of an ISO standard without making it too time consuming and labor intensive?

Overcoming FP Challenges — Early and Quick High Level Project Sizing Using FP Sampling

One way to overcome the challenges of using FP described above is to use it “behind the scenes” on a limited basis to do FP sampling of countable artifacts (e.g. use cases, user stories, etc.) to derive a gearing or conversion

factor. The Software Sizing Infographic by QSM provides a list of the most common artifacts that can be normalized to FP (see sidebar.) This helps overcome resistance to function points by describing functionality in units a given organization and culture can understand (e.g. agile user stories) that can be translated into estimated FP behind the scenes as an input to a parametric estimating tool such as SLIM® or COCOMO®.

Early estimates are needed long before a project even becomes a project, when it is still an idea or concept in the making. Executives have discussions about affordability (is it within the realm of possibility in terms of cost,) resources (do we have the right people to even attempt it) and schedule (what will we have to set aside and for how long to get this done?) At this point in time, little is known aside from preliminary functions (it's going to revolutionize customer service, for example) and certainly not enough to do a detailed FP count. As figure 1 below indicates, the cone of uncertainty is high when you are early in the software development life cycle (SDLC). Nonetheless, an estimate is often needed to support bids and corporate planning. This is where using gearing factors to approximate the number of FP based on a count of available artifacts can provide value.



The cone of uncertainty (fig 1) is a widely accepted concept and can be used to set expectations with

stakeholders for estimates performed at various stages of the SDLC. At each stage of the SDLC, FP sampling can be used to establish gearing factors for whatever requirements artifacts are available at that point in time.

The benefits of a sampling approach to FP counting include:

1. Less time consuming and labor intensive (a representative sample of high level requirements are taken);
2. Can be done “behind the scenes” in organizations that are adverse to, or do not understand, FP counting and the resultant numbers are expressed in more acceptable units-of-measure and
3. Usually there is something countable like business requirements, use cases or user stories where you can establish a gearing factor (i.e. ratio).

How big should the sample size be? Even a small sample size can provide some value in approximating the gearing factor for various artifacts. However, for a more robust gearing factor, expert statistician and Certified Six Sigma Black Belt (CSSBB) Paul Below recommends a sample size of at least 12.

Let’s include an example here. The goal in this example is to use FP sampling to establish gearing factors for both use cases and user stories for Company X who develops educational software for colleges and universities. The Course Registration System project is considered a representative example of the types of projects developed by company X. A sample of 8 use cases and 35 user stories is chosen. Company X defines a user story as a thread of functionality within a use case (a.k.a. use case scenario or flow). The details of the use cases and user stories are based on the Course Registration System case study on page 483 of the IT Measurement Compendium. The FP counts were performed by an IFPUG Certified Function Point Specialist (CFPS).

Project Name	Use Cases	User Stories	IFPUG Function Points	User Stories per Use Case	FP per User Story	FP per Use Case
Course Registration System	8	35	108	4	3	14

Now that we have some gearing factors, we can quickly ballpark the number of function points for future estimates and analyses. For example, if you are describing a new project as “about 20 use cases,” you know that it is approximately 280 function points. You can then input that size assumption (280 function points), along with size uncertainty (based on where you are in the SDLC vs. the cone of uncertainty) into a parametric tool like SLIM® to determine the feasibility of developing and delivering that functionality within a given budget and schedule.

Using the FP sampling approach is one way to improve how we size projects and increase the robustness of our software estimates. The better are the estimates, the better will be the on-time, on-budget, and with a satisfactory result, project successes.

References

1. Standish Group 2015 Chaos Report - Q&A with Jennifer Lynch, https://www.infoq.com/ articles/standish-chaos-2015
2. Standish Group 2015 Chaos Report - Q&A with Jennifer Lynch, https://www.infoq.com/ articles/standish-

chaos-2015
3. Quantitative Software Management (QSM) Software Sizing Infographic outlines how and when to use functional size measurement during the software development life cycle. Both authors were involved in the development of this important infographic. - https://www.qsm.com/infographic/software-sizing-matters
4. ISO/IEC 20926:2009 IFPUG 4.3.1 Functional Size Measurement Method, http://www.iso.org/iso/catalogue_detail. htm?csnumber=51717
5. On XML Based Automated Function Point Analysis: An Effective Method to Assess Developer Productivity Jeffrey S. Lent and Yanzhen Qu, Lecture Notes on Software Engineering, Vol. 3, No. 4, November 2015, (http://www.lnse.org/vol3/199-X0016.pdf)
6. Cone of Uncertainty from the QSM Software Sizing Infographic: https://www.qsm.com/infographic/software-sizing-matters
7. Sample Sizes and Trend Line Creation Paul Below, 2016 QSM Software Almanac, page 37 (https://www.qsm.com/resources/software-almanac-2016)
8. The IT Measurement Compendium: Estimating and Benchmarking Success with Functional Size Measurement, Manfred Bundschuh and Carol Dekkers, 2008, https://www.amazon.com/exec/obidos/ASIN/3540681876/qualitplustec-20