![DXC Technology logo]

How can you be sure you're getting full development value for money?

by **Mark Reynolds**, Director, Banking and Capital Markets Solutions, DXC Luxoft

and **Balaji Venkatramani**, Head of APAC Delivery Strategy and Solutions, DXC Luxoft

![Luxoft - A DXC Technology Company logo]

When pressed, more enlightened decision-makers say they regard technology as a business and innovation partner. But all too often, the technology division is seen as one dimensional; simply accountable to their business stakeholders for work done and change delivered within the parameters of a given budget.

It's a simple enough equation: Output divided by input. And a successful solution delivers the greatest output for the least cost.

But it's not just about the volume of change delivered — producing broken code in an unreliable timeframe is unlikely to win the authors many friends. No, it's also about the *quality* of change, delivering working code through to production without breaking or destabilizing the existing system. Another vital characteristic is *predictability*, delivering the agreed scope on time, every time.

All or nothing

Like football, it's a team game; you *win together and lose together*. And, as in the *beautiful game*, recruiting a group of high-performing individuals doesn't guarantee high-level output. Your team has to function as a unit, with senior developers mentoring less experienced colleagues. Consequently, you need to measure the output of the whole team rather than individual team members.

Most software delivery metrics focus on coding, but this is only one part of the process. Unclear or unstable requirements and poor testing can result in a drastic reduction in output, so look at the whole picture — the total delivery lifecycle.

Measuring the right output

While it's relatively easy to measure output in isolation, it's virtually impossible to measure whole systems or teams — the input needed to produce it will vary greatly between systems. Output will differ significantly according to whether you're working on an old, undocumented monolithic application with no testing or a brand new unit-tested set of microservices.

Whichever metrics you use, at least you can compare them at specific times to get an idea how your teams are performing and whether the trend is positive.

Factors that affect the rate of change:
- Age and complexity of system and codebase
- Type and scope of the change
- Documentational quality
- Greenfield versus brownfield project
- Architectural complexity and flexibility (e.g., monolithic versus microservices)
- Tools and technologies used
- Automation levels enabled
- Software development lifecycle (SDLC) and release cycle
- Test environment availability
- Business requirement quality and stability

Criteria

Software development's greatest attribute is that everything is measurable. And as a delivery organization matures, an increasing number of items can be tracked via a dashboard, allowing the regular monitoring of trends. Items like:

Output

- **Stories delivered** — a guide to business value
- **Pull requests per sprint** — incremental delivery of value into test environments

Complexity

- **Size of the codebase** — the magnitude of one system versus another
- **Coding style violations** — how much of the code follows best practice for maintainability

Quality

- **Defects found in each phase of the lifecycle** — how many bugs are finding their way through each stage of testing and into production
- **Capacity allocated to new functionality versus bug fixing** — the effort spent retracing your footsteps

Predictability

- **Actual versus estimated effort** — understand and refine estimate accuracy, learning from experience
- **Average delivery cycle time** — how long it takes for a feature or user story to go from requirements specification to production

Process maturity

- **Knowledge index** — spread of expertise across the team and different components
- **Requirements stability index** — reliability of incoming requirements

Production stability

- **SME callouts** — out-of-hours, critical issue investigations
- **Average time to resolve production issues** — how soon the system can be fixed

Team stability

- **Employee motivation index** — signs that your team is happy at work
- **Attrition** — level of employee turnover (or churn) in the organization

Interpret your measurements

Measuring teams is fine as far as it goes, but what do the measurements actually tell you? Correlation between metrics can strengthen the signals and flag-up when it's time for action.

Potential risks you could diagnose:

Issues with the clarity of requirements:

- Lengthening backlog queue
- Diminishing team velocity
- Fewer user stories accepted

Release date at risk:

- Behind schedule in project burn up/burn down
- Reduced *defect phase containment* — bugs not picked up

Unplanned scope or overworked team:

- Behind schedule for project burn up
- More user stories accepted

Growing complexity of features, stories or architecture:

- Diminishing team velocity
- Extra Coding Effort (as measured by BlueOptima)
- Increase in average delivery cycle time

What drives output creation?

The scale of software engineering output depends on a combination of many factors, including individual excellence, teamwork and the company environment.

Clearly, you need to hire the best engineers you can find. At DXC Luxoft, we have a large presence in parts of the world that give us easy access to top talent.

As well as technical skills, you want individuals with the right project experience and domain knowledge — people who have worked in your industry before and understand the nuances of your particular project environments.

Once you have the raw material, it's about the way you set up and operate your teams. As mentioned earlier, just having a group of smart developers is not enough. Success relies on getting them to function as a cohesive, high-performance unit. This involves delivery processes and tooling, as well as best practices such as ensuring efficient knowledge management to avoid key person dependency.

Then there's the ex-factor — building a culture of excellent performance, with exemplary training and exceptional motivation. HR processes have a central role to play here, promoting diversity and inclusivity to create a stronger team ethic and a more eurythmic

workforce. The whole is stronger than the sum of its parts, and engendering a sense of belonging and respect, alongside peer and corporate recognition is a tried and tested way of developing and retaining your best talent. After all, staff retention is a key factor in ensuring predictable delivery.

With all that in mind, we must counsel you against assessing your development organization on a cost-per-head basis. Team and company considerations can have a telling impact on overall performance.

The company		
The team		
	The individual	

The company

- Company culture
- Employee training
- Working facilities
- Employee motivation
- Time to ramp up
- Level of attrition

The team

- Team configuration
- Knowledge management
- Delivery processes
- Tooling
- Automation

The individual

- Aptitude
- Level and range of technical skills
- Project experience
- Domain knowledge
- Experience in the particular environment
- Location and timezone proximity to colleagues and stakeholders
- Communication and English skills

To misquote John Donne, *No manager is an island*, and the technology division is not *entire of itself.* In other words, no tech team performs in isolation.

Healthy collaboration between both individual technology teams, and technology and the business is a central pillar of software development success.

However, technology teams can only fully engage with the business if they have sufficient domain knowledge to do so (which, we're pleased to report, DXC Luxoft teams do).

As you can see, getting optimum value for money from development teams isn't just about cutting costs — nowhere near.

There are a great many collaborative ducks of all shapes and sizes to get in a row before all parties can say they're truly aligned.

Get in touch

Firms across Capital Markets need a dependable partner they can rely on to offer guidance, expertise and technical assistance at every step of the project journey.

DXC Luxoft expert delivery teams can help clients prioritize and execute on projects that have a *direct* impact on the bottom line, to empower firms to optimize their use of capital for improved profitability.
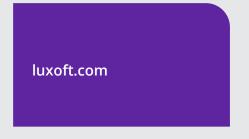

Mark Reynolds, Head of BCM Portfolio Marketing

Mark has 14 years' experience in Financial Services technology, enabling strategy and business growth in a variety of client-facing leadership and business development roles. He specializes in working with nearshore technology delivery organizations, having built and led teams across central and Eastern Europe.

Balaji Venkatramani, APAC Delivery Strategy and Solutions Head

Balaji Kumar is a senior director with DXC Luxoft India, and heads the Digital Delivery Strategy and Solutions for BCM APAC.
He has over 21 years' experience in the IT industry. Balaji has driven large-scale technology solutions and transformation initiatives, directly, in Silicon Valley technology companies as well as in services partnership with global financial clients. He specializes in large-scale knowledge transitions, transformations, Agile, DevOps, big data and analytics, cloud and program management.

luxoft.com

About DXC Luxoft

DXC Luxoft is the design, data and development arm of DXC Technology, providing bespoke, end-to-end technology solutions for mission-critical systems, products and services. We help create data-fueled organizations, solving complex operational, technological and strategic challenges. Our passion is building resilient businesses, while generating new business channels and revenue streams, exceptional user experiences and modernized operations at scale.