



Boosting Agile Velocity with Factory as a Service on Azure

Jean-Louis Lalonde, Eng., M.Eng.
President, CEO
AZUR Group
April 2015
Montreal, Canada

EXECUTIVE SUMMARY

Our experience has proven it time and time again: Organizations that migrate to the Cloud save on IT spending. These savings come not only from a reduction in data center expenses, but also from lower IT personnel costs. Because Cloud-based systems reduce the effort required for ongoing support, Cloud users are able to devote a higher percentage of their IT spending to new initiatives. The cost savings, combined with strategic benefits in speed, scalability and agility, are strong arguments for any organization to take a more aggressive leap to into the Cloud.

The obstacle facing many organizations today is how to phase out antiquated legacy systems and revitalize the business with a *true Cloud upgrade that harnesses the power of the Cloud* — not just a migration. Companies need to achieve this without spending disproportionate funds on these projects. To get them there, they need a faster Agile software development process — one that will take their commerce to a more competitive place in the market in much less time.

Enter **Factory as a Service** (FaaS) — AZUR Group's unique innovation, which achieves this sought-after boost to Agile velocity. FaaS delivers the automation of some of the most tedious, time consuming and resource-gouging elements of Agile that will change the way companies do the business of software development for the Microsoft Azure Cloud platform.

Case in point: A medium-complexity Agile project over 24 weeks (8 sprints). XI-Factory™ FaaS implementation automatically generated some 93,033 lines of code (LOC) versus 20,978 manually developed LOC. That includes an average of 6 times more story points than those developed manually (1630/264). If these story points had not been automatically generated using bug-free, high quality code templates written by our experts, the team would have had to code the story points themselves — adding many months to the project timeline. Contact us to see how we can apply these figures to your project.

SOFTWARE MODERNIZATION TO CLOUD SERVICES

As in-house legacy systems meet their imminent demise, the business of Software as a Service (SaaS) is gaining more and more momentum. SaaS is expected to deliver one quarter of all software revenues by 2016.

It's time for organizations to get on board with the Cloud model. The industry is currently overflowing with millions of in-house legacy systems that are reaching the end of their life cycle. This includes old desktop applications from an assortment of generic technologies, such as client-server applications, spreadsheets and database management systems. These intangible assets make it difficult for competitors to emulate and give them a competitive advantage as key differentiators. Furthermore, companies are finding that commercial off-the-shelf (COTS) software packages are not always the best solution for maintaining that unique competitive edge.

These legacy systems were built with significant capital investments over time. But regardless of their real or perceived value, they must be renewed into the Cloud and mobile formats in order for the business to remain current and fulfill today's demanding customer needs. But that transformation does not come without its obstacles. Here are some of the issues around software development for the Cloud and mobile platforms:

-
- **A shortage of proficient, affordable developers**
 - **A lack of documentation explaining Cloud limitations**
 - **The intensity and costliness of R&D cycles required prior to Cloud transformation**
 - **New software development and software modernization projects involving excessive analysis, design, development and QA cycles. This — combined with poor time-to-market and prohibitive development costs — result in mass organizational slowdowns.**
 - **The complexity of business requirements is exploding, resulting in more of the above resources and costs.**
 - **Legacy systems are built using a mix of technology like DB2, Sybase, Oracle, Access, FoxPro, Excel, Delphi, VB, COBOL, Novell, and Lotus Notes, among others.**

THE EVOLUTION OF SOFTWARE DEVELOPMENT METHODOLOGY

How can we become faster at building new software for the Cloud to keep up with worldwide demand?

Software Development Methodology's goal is to divide development work into distinct phases with corresponding activities, with an eye to streamlining planning and management. These methodologies have evolved from waterfall, spiral development, rapid application development and now Agile methodology. Agile is a popular process for Cloud software development, thanks to its flexible and iterative ability to deliver enterprise value that's faster, reduces project risks and adapts to changes in a timelier manner.

Agile methodology delivers a flexible approach to the direction of the software development lifecycle. This is achieved through regular paces of work, known as sprints, following which teams must present a complete iteration. Agile's repeated, condensed work cycles — along with the products it bears — are what make Agile methodology incremental in nature. With waterfall methodology, development teams only have one opportunity to get it right. In the Agile world, every piece of development — requirements, design, etc. — is continually reassessed. There's always that chance to re-evaluate, re-assess, and change direction.

This flexible approach to development greatly reduces costs and time-to-market. Because teams can develop software and gather requirements simultaneously, analysis paralysis is less likely to set in and hinder progress. The two-week work cycle presents stakeholders the opportunity to adjust releases based on emerging, real-world business needs.

THE EVOLUTION OF CLOUD COMPUTING

One of the first milestones of Cloud computing was the arrival of Salesforce.com in 1999 as a Software as a Service, which pioneered the concept of delivering enterprise applications via a simple website. This service business paved the way for both specialist and mainstream software firms to deliver applications over the Internet.

The next breakthrough was in 2002, when Amazon Web Services provided a suite of Cloud-based services including storage and computation. This was followed by other major players like Microsoft Azure in 2010, and Google Cloud in 2012, among others. The objectives of Cloud Computing are to reduce investments and proportional IT costs while increasing scalability, availability and reliability.

The diagram below depicts the Cloud Computing stack, displaying three distinct categories within Cloud computing:

- **SOFTWARE AS A SERVICE (SaaS)** is a Cloud-based application designed for end-users and delivered over the Internet.
- **PLATFORM AS A SERVICE (PaaS)** is the set of tools and services designed to make coding and deploying those applications quick and efficient. By providing the underlying software infrastructure, PaaS can reduce organizational costs. PaaS adds velocity, flexibility and agility to software development. It creates a reliable, diverse application infrastructure that won't obstruct organizations from quickly meeting their customers' needs.
- **INFRASTRUCTURE AS A SERVICE (IaaS)** is the hardware and software that powers it all: servers, storage, networks and operating systems. It lowers hardware costs; provides almost unlimited scalability and agility, and a faster time-to-market — with superior availability, security and compliance. The leading incentives for using IaaS are: hardware and infrastructure cost savings; capacity management; disaster recovery & business continuity; IT staffing & administration cost savings and access to new expertise and abilities.

Software as a service

Platform as a service

Infrastructure as a service

Fig. 1: Cloud Computing Stack

THE NEW FaaS SOLUTION

How can we improve software development and deployment with automated Cloud mechanisms to gain higher productivity, consistency and quality?

Despite Agile's best intentions, Agile-developed software for Cloud computing platforms is still a highly manual process with considerable human intervention required at every phase.

When it comes to a software development project, time is always of the essence. That's because of the constant evolution of the way organizations do business, collaborate within other organizations and customers, and modify internal business processes. Developing faster and better applications is still a pain point for organizations that have not been able to speed the development process and that experience lags between requirements gathering and application artifact delivery.

So although the Agile process serves IT and business analysts when it comes to requirements gathering, the quality of user stories is often lacking. Moreover, the software coding process is not nearly as quick to keep up with the fast paced IT business analysts.

What we need is a new approach to software construction and modernization. And it needs to be an efficient, collaborative and automated software development process.

ADVANTAGES OF AUTOMATION:

The best reason to automate is to benefit from higher production rates and increased productivity. Automation also delivers better product quality, improved security, shorter workweeks for labor, and reduced factory lead times.

FLEXIBILITY:

Tasks that require manual dexterity, like programming multifaceted business rules from complex designs, will always require the intellectual interventions of people. The goal is not to automate all software development activities, but rather to automate the easiest and most tedious programming tasks. Given that the suitable code templates for user services, business services and data services are provided; automatic generation is well within reach.

COSTS:

While automation can lower variable costs over time, the initial development costs can be prohibitive. It's notoriously difficult to accurately estimate the research and development costs required to design the machinery to automate processes. If the company doesn't have adequate cash flow and expertise, the automation option may be off the table. This is why most companies should use ready-made, affordable automated Cloud services to achieve efficient, custom software development.

FACTORY AS A SERVICE (FaaS) IMPROVES THE EFFICIENCY OF THE SOFTWARE MODERNIZATION PROCESS FOR THE CLOUD AS FOLLOWS:

HIGHER QUALITY PROCESS INPUTS:

- Automatic data model importation from old software database
- Higher quality software user stories resulting from a better online collaborative process
- Pre-built software template structures ready for the intricacies of Azure IaaS and PaaS architectures

QUICKER DEVELOPMENT PROCESS

- SaaS design from a simple online configuration process
- Both online and offline phases of code generation for all software tiers to jump start development cycles
- Automatic SaaS deployment on Azure IaaS and PaaS

HIGHER AVAILABILITY OF PROCESS OUTPUT:

- Intermediate SaaS version can be quickly and securely tested from anywhere by project stakeholders for invaluable feedback.

- **FACTORY AS A SERVICE (FaaS)** is the set of tools and services designed to support a high velocity Agile software development process from increased process automation with automatic SaaS code generation capability.

The diagram below depicts the improved Cloud Computing stack. It shows four distinct categories within Cloud Computing:

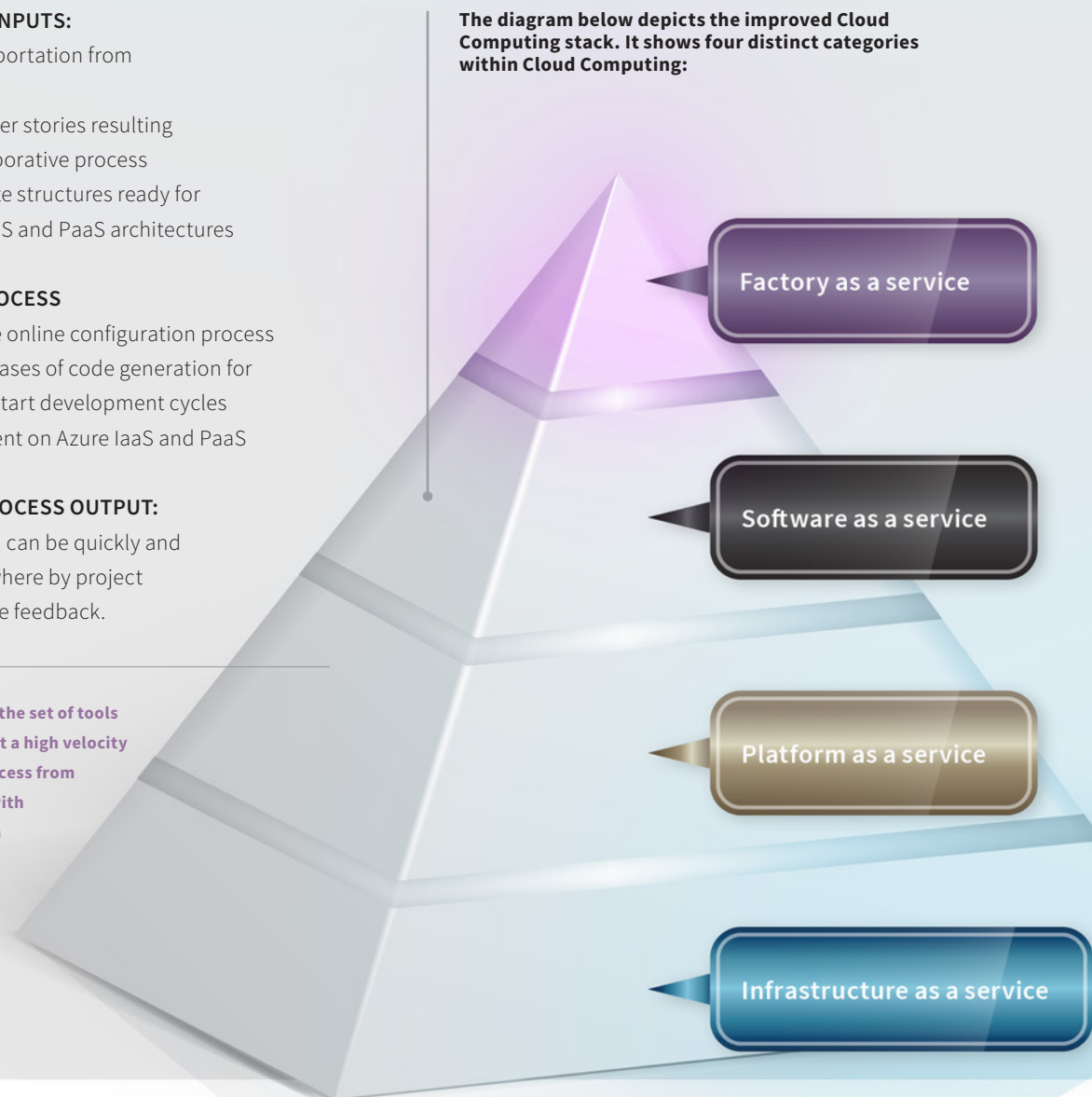


Fig. 2: FaaS + Cloud Computing (SaaS, PaaS, IaaS)

FACTORY AS A SERVICE IMPLEMENTATION

XI-Factory™ is the AZUR Group implementation of a Factory as a Service for the Microsoft Azure Cloud platform, supporting an Agile online process with increased velocity.

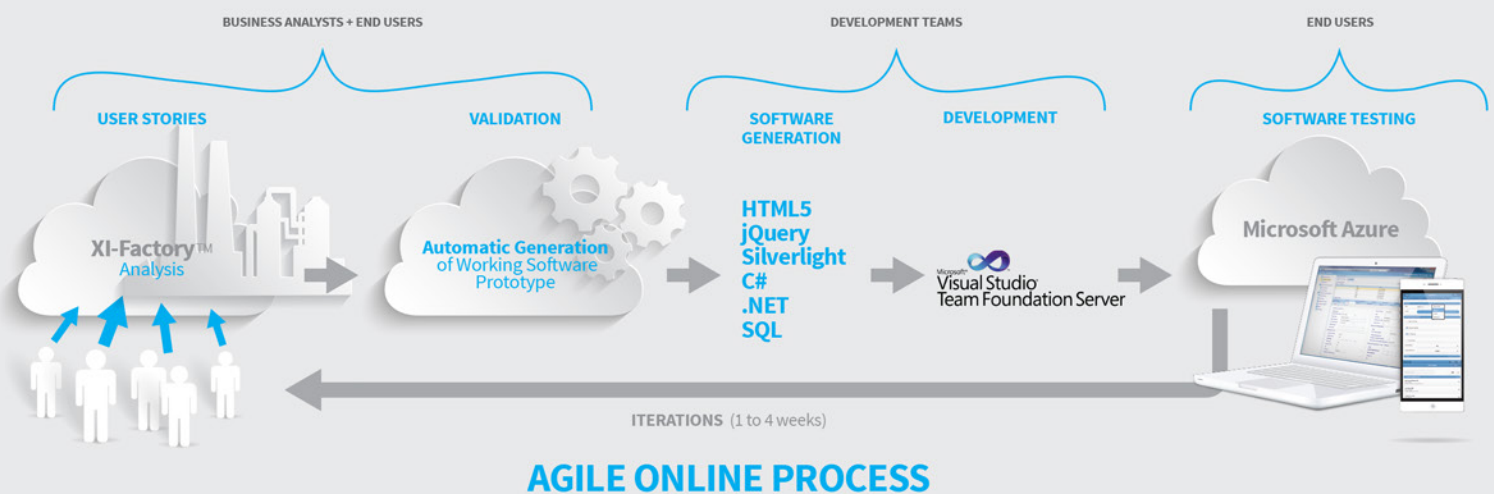


Fig. 3: XI-Factory's Agile Online Process

XI-Factory's AGILE ONLINE PROCESS MAIN STEPS:

1. Define sprint backlog
2. Collaborate with team members to write and validate user stories
3. Design and configure software data and functional models
 - Use automatic importation tool to retrieve data models from existing database systems
4. Use the first online code generator to get a working software prototype; obtain feedback and consent from stakeholders
5. Once approved, use the second offline code generator to generate the solution code for the developer teams.
6. Add the code and the user stories to the Microsoft Team Foundation Server (TFS)
7. Developers use Microsoft Visual Studio to add business rules to the generated code
8. Deploy intermediate SaaS version on Azure and have stakeholders test it online
9. Return to step 1

FACTORY AS A SERVICE IMPLEMENTATION

With the XI-Factory design interface shown in Figure 4, your IT business analyst can design and configure a SaaS solution, (like the Order and Invoice Management depicted here) with the data and functional models in the left tree pane. User stories can be entered for each software definition leaf and can be revised by stakeholders.

At any moment, the IT business analyst can initiate the SaaS solution code generation process to visualize the desktop and mobile software prototypes.

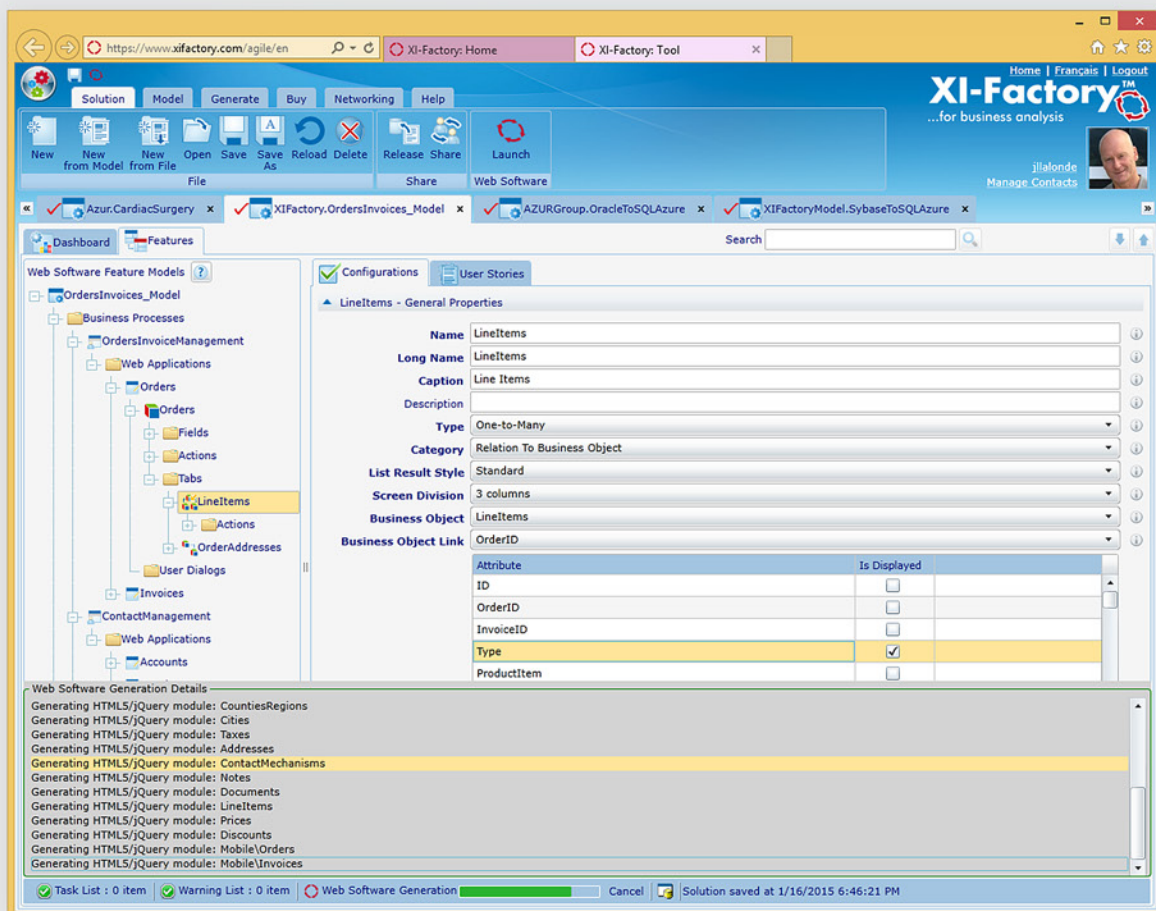


Fig. 4: XI-Factory Tool - Main Design Interface

FACTORY AS A SERVICE IMPLEMENTATION

Business Analysts can present the newly-generated, working desktop and mobile software prototypes to stakeholders to get their feedback.

For example, this Order and Invoice management model solution has some 74,275 lines of code (LOC); all automatically generated with XI-Factory in just a few seconds.

Developers can later add user story business rules with Visual Studio to complete solution functionalities.

ORDER AND INVOICE MANAGEMENT SOLUTION SCOPE:

Business Process: 6; Web Application: 19; Field: 783; Action: 458; Relational Table: 34

FaaS AUTOMATIC LOC GENERATION:

**User Services: 65,573 LOC; Business Services: 1,814 LOC; Data Access Services: 2,631 LOC; Entity Framework Services: 4,257 LOC
Total: 74,275 LOC**

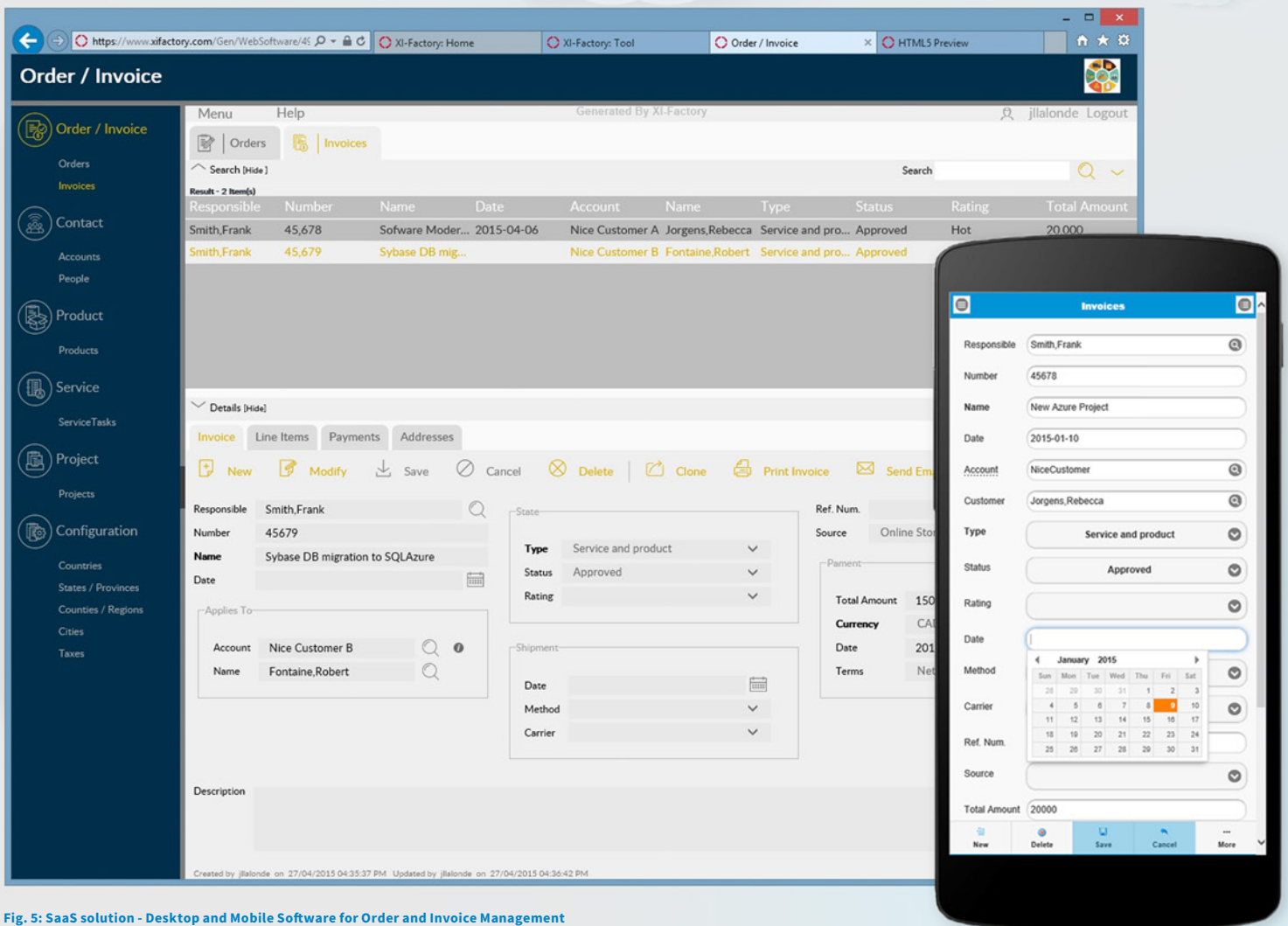


Fig. 5: SaaS solution - Desktop and Mobile Software for Order and Invoice Management

FACTORY AS A SERVICE IMPLEMENTATION

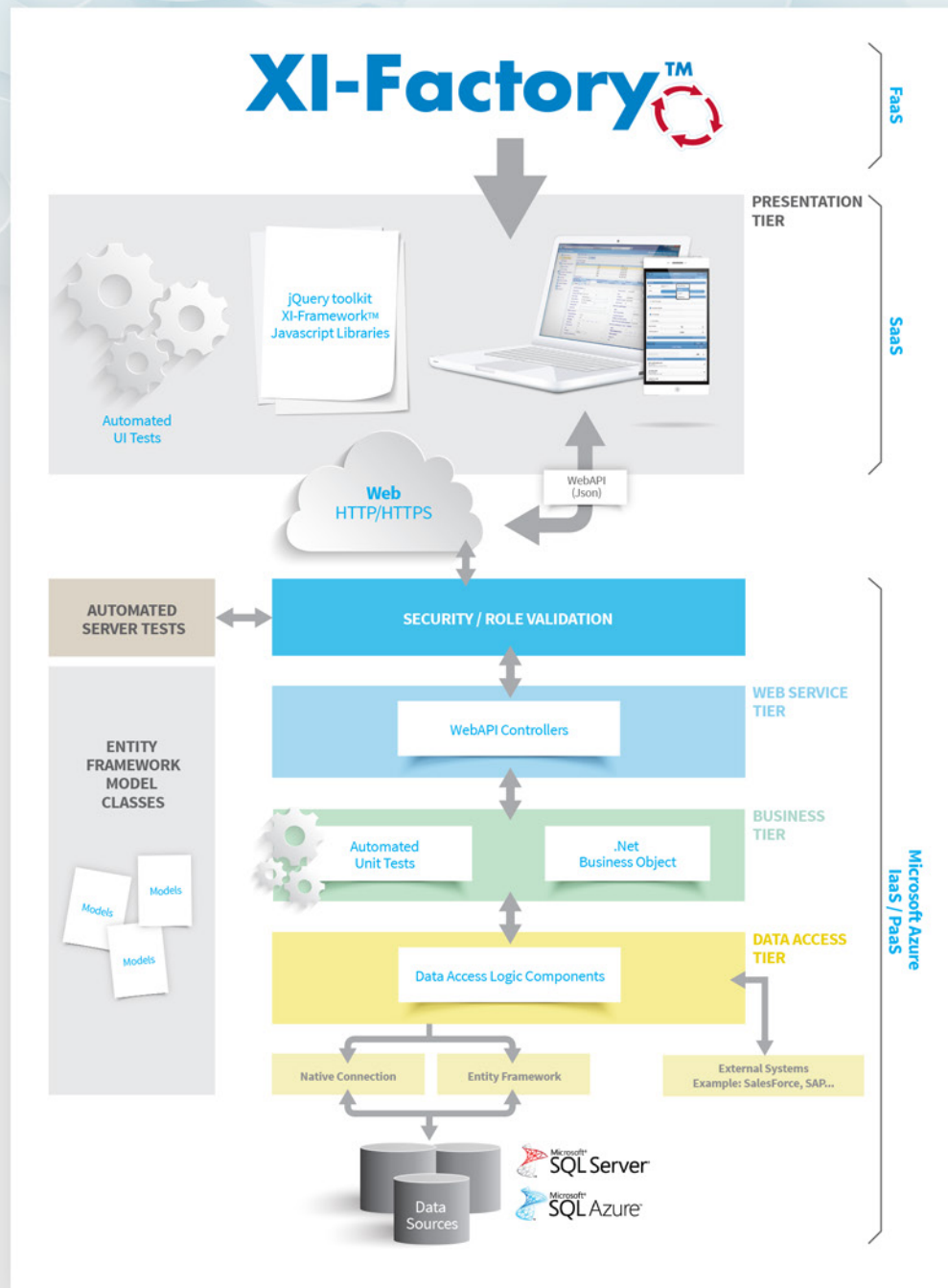


Fig. 6: Cloud-Based Software Architecture

RESULTS:

The following Agile concepts illustrate the efficiency of FaaS in terms of improved speed and quality.

USER STORY DEFINITION:

In consultation with the customer or product owner, the team divides up the work into functional increments called user stories. Each user story is expected to yield, once implemented, a contribution to the value of the overall software, irrespective of the order of implementation.

USER STORY POINTS:

A way to define the user stories' complexity relationships; with points representing the amount of effort involved in delivering the story.

VELOCITY DEFINITION:

At the end of each iteration, the team adds up effort estimates associated with user stories completed during that iteration. This total is called velocity. Calculating the velocity enables the team to compute (or revise) an estimate of the project duration based on the estimates associated with remaining user stories—and assume whether the velocity will remain approximately the same for the remaining iterations. This is generally an accurate prediction, but rarely a precise one.

Velocity is a metric that predicts how much work an Agile software development team can successfully complete within a two-week sprint (or similar time-boxed period). For example, if the team completed 10 stories during a two-week sprint and each story was worth 3 story points, then the team's velocity is 30 story points per sprint. To demonstrate FaaS velocity using XI-Factory, we are using line of code (LOC) to appreciate its capability. User stories are defined with story points, which can be evaluated with LOC, i.e., developed manually and automatically generated, resulting in the following relationship:

-
- **User story = Number of Story points = LOC_Dev (Number Line of code developed manually) + LOC_Gen (Number Line of code generated automatically):**

- LOC (Physical lines). This is the simplest line count. Each line ends with a line break, usually CR+LF. LINES counts every line, be it a code, a comment or an empty line.
- Types of line of code for N-Tier software architecture: HTML5, jQuery, Jason, Rasor, C#, SQL

RESULTS:

As presented in Table 1, (a medium complexity project designed with XI-Factory), our team was able to automatically generate — in a few minutes — an average of 6 times more story points than were developed (1630/264) over 8 sprints. If these story points had not been automatically

generated from bug-free, high quality code templates written by our experts, the team would have been required to code the story points — adding months to the project timeline.

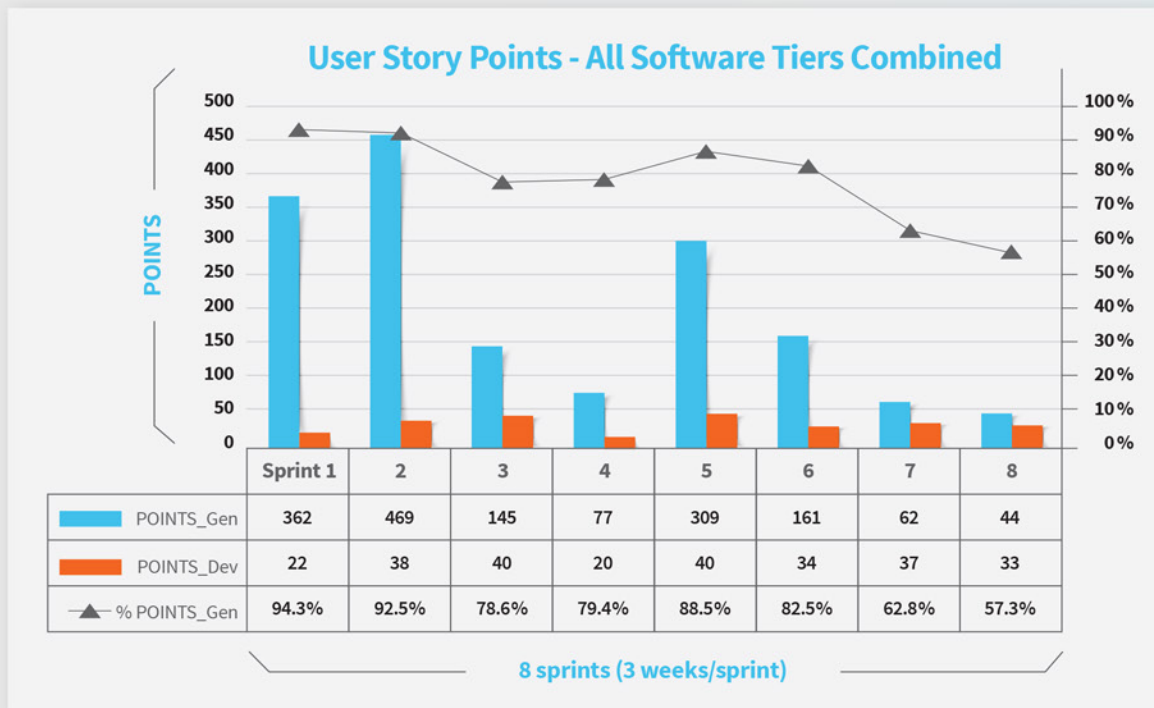
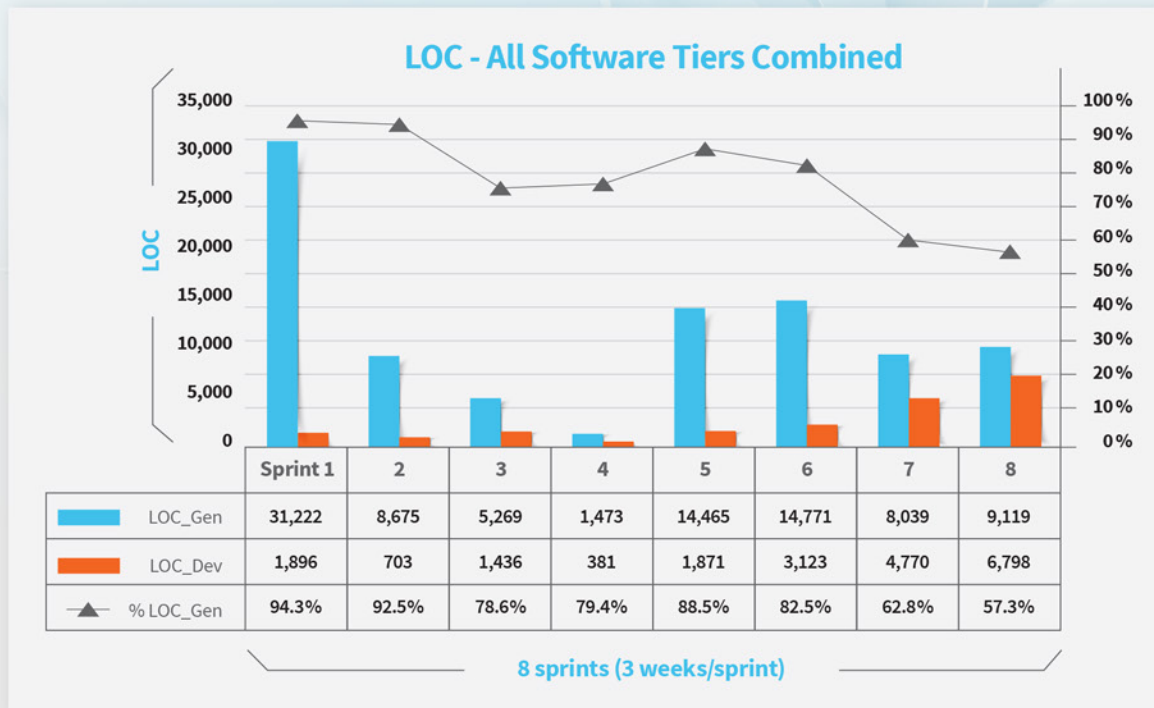
Software Scope per Print	1	2	3	4	5	6	7	8	TOTAL
Developer	6	6	7	4	6	6	8	6.5	
Story Point Generated	362	469	145	77	309	161	62	44	1630
Story Point Developed	22	38	40	20	40	34	37	33	264
% Generated	94.3%	92.5%	78.6%	79.4%	88.5%	82.5%	62.8%	57.3%	86.08%
Business Process	1	2	1	0	0	0	0	1	5
Web Application	10	6	6	0	6	2	2	4	36
Field	197	167	105	46	375	225	163	282	1560
Action	124	78	78	10	224	131	39	195	879
Relational Table	19	9	14	6	8	13	4	11	84

8 sprints of 3 weeks each

Table 1: Agile data from a medium complexity project

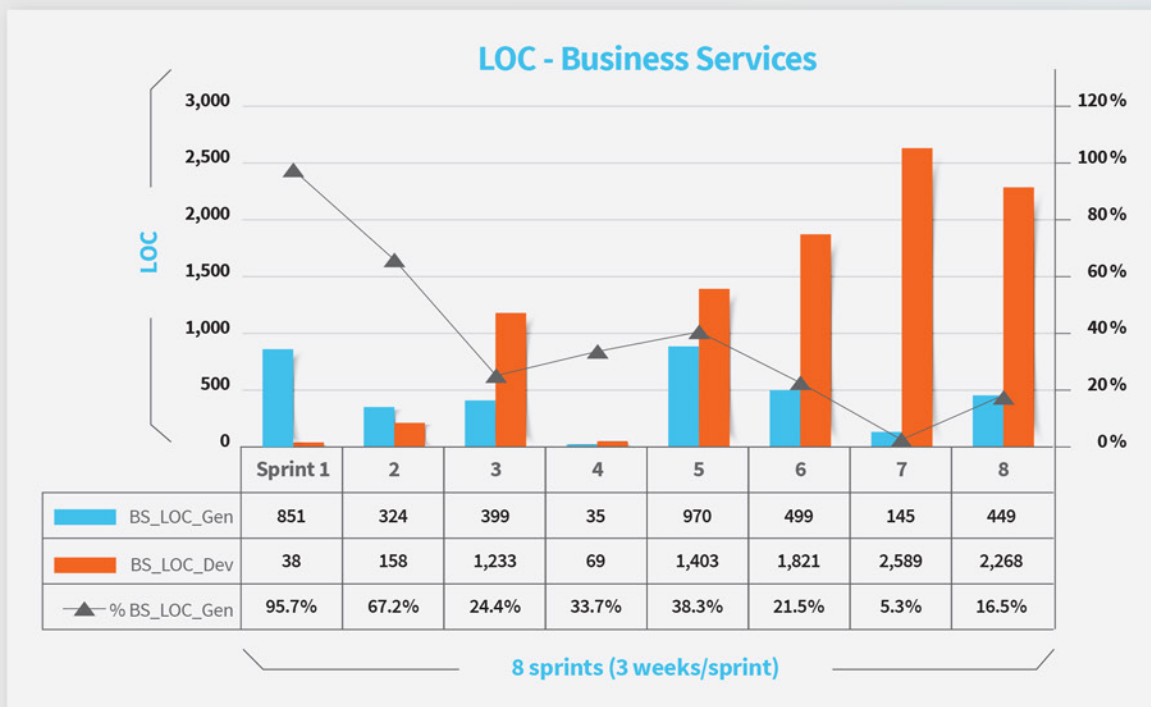
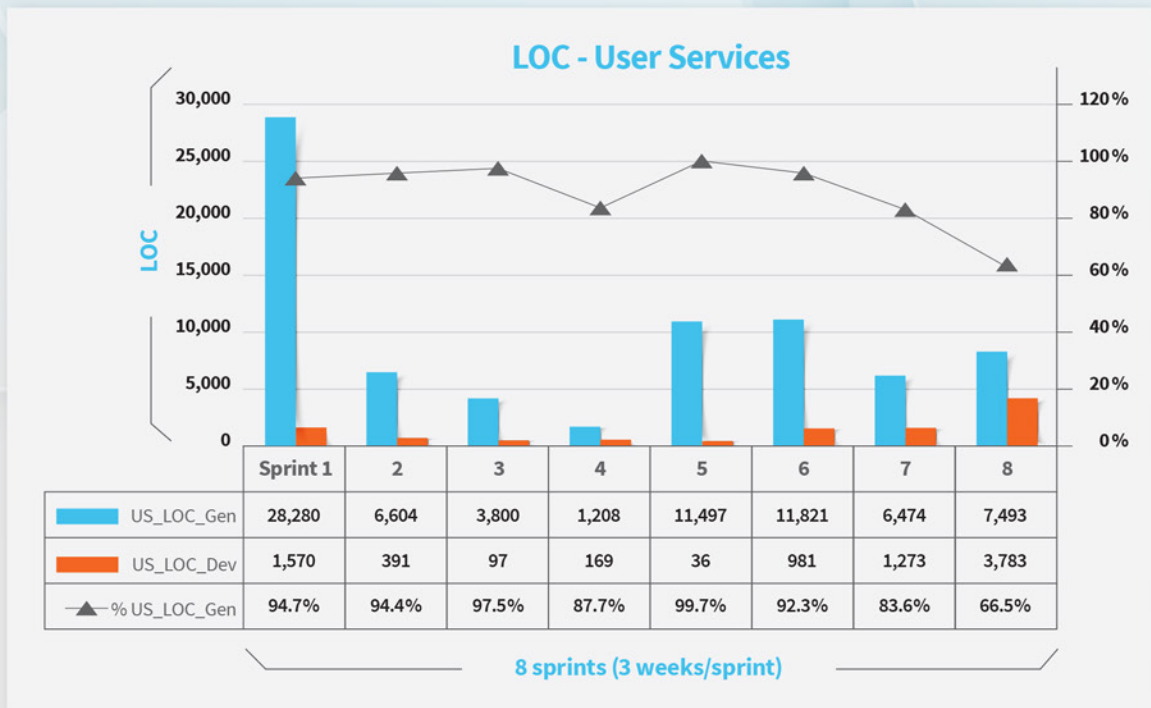
RESULTS:

GENERATED VS DEVELOPED The two diagrams below depict the FaaS velocity in terms of User Story Points or LOC developed manually by the team and generated automatically with XI-Factory for each sprint.



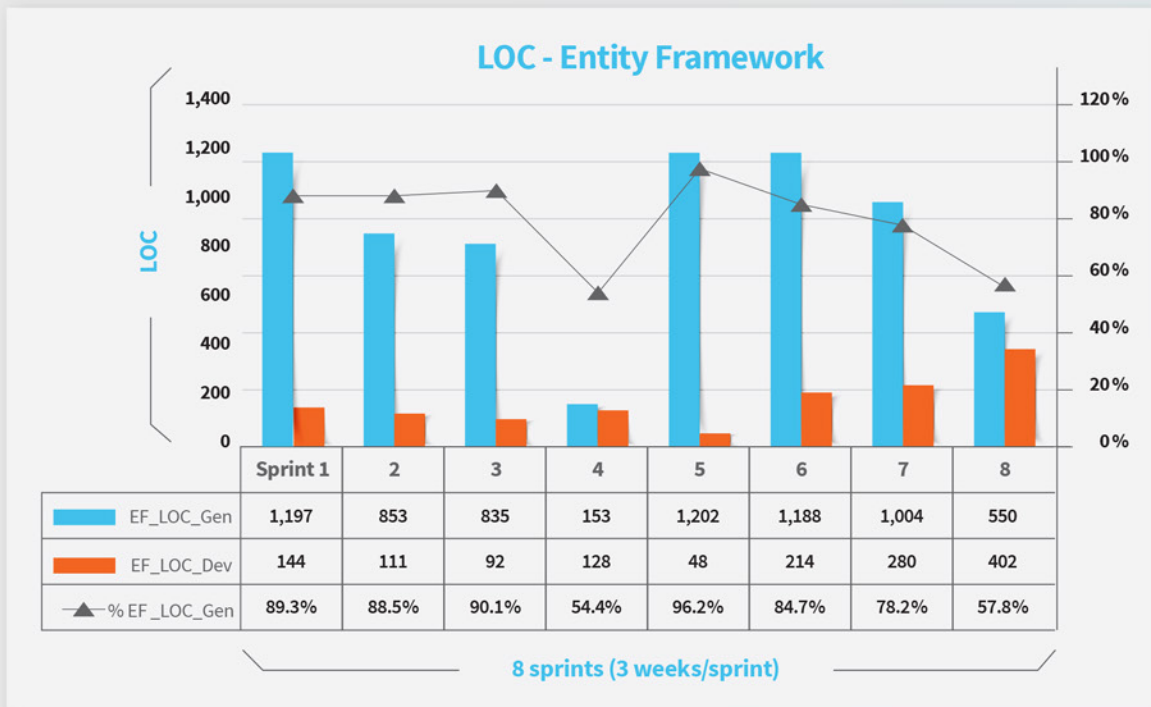
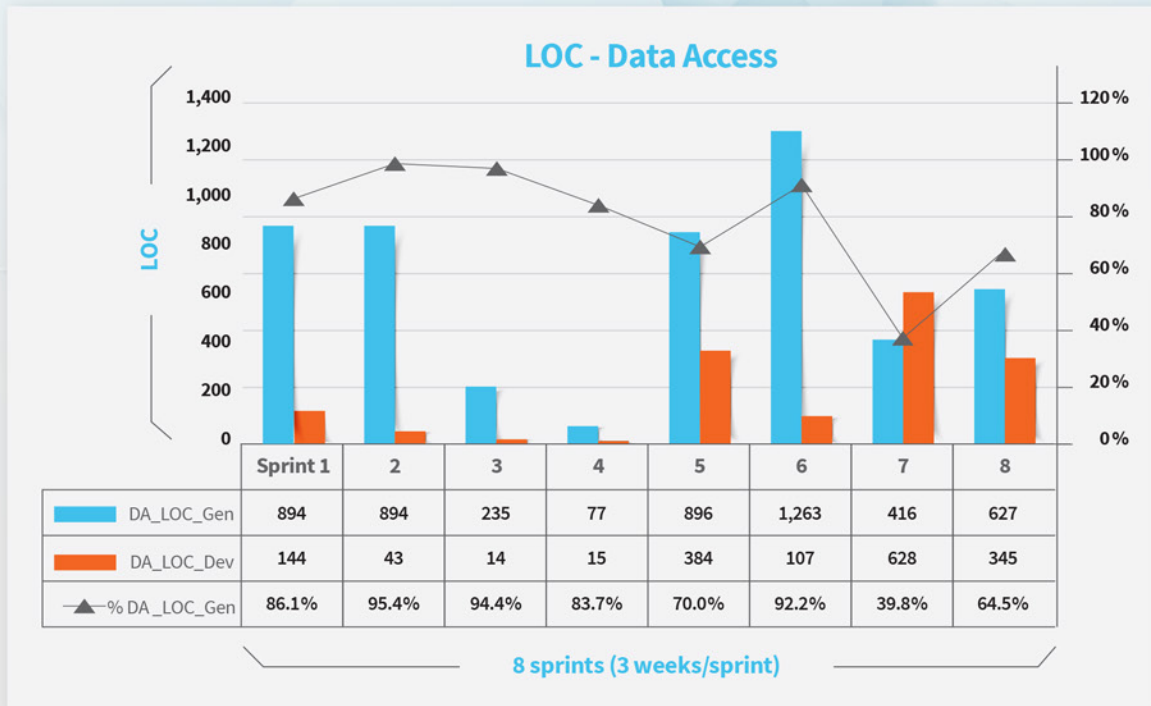
RESULTS:

GENERATED VS DEVELOPED The four diagrams that follow depict the FaaS velocity in terms of LOC obtained manually and automatically for each sprint including: **User Services**, **Business Services**, **Data Access Services**, and **Entity Framework Services**.



RESULTS:

GENERATED VS DEVELOPED



CONCLUSION

It's time to take Agile methodology velocity to the next level with FaaS — and strip it of as many manual processes as possible. What we're left with is a lean, rapid development process that enables you to achieve your business and IT goals, decreases your time-to-market, and helps you reach the Cloud faster — at a cost your organization can afford.

XI-Factory automation delivers more than just a more efficient, affordable and streamlined Agile software development process. It also provides the opportunity to alleviate the pressure of an important group in your organization — namely developers— of repetitive, unpleasant work. It opens the door to future automation technologies to provide a growing social and economic environment in which we can enjoy a higher standard of living and a better way of life.

ABOUT AZUR GROUP

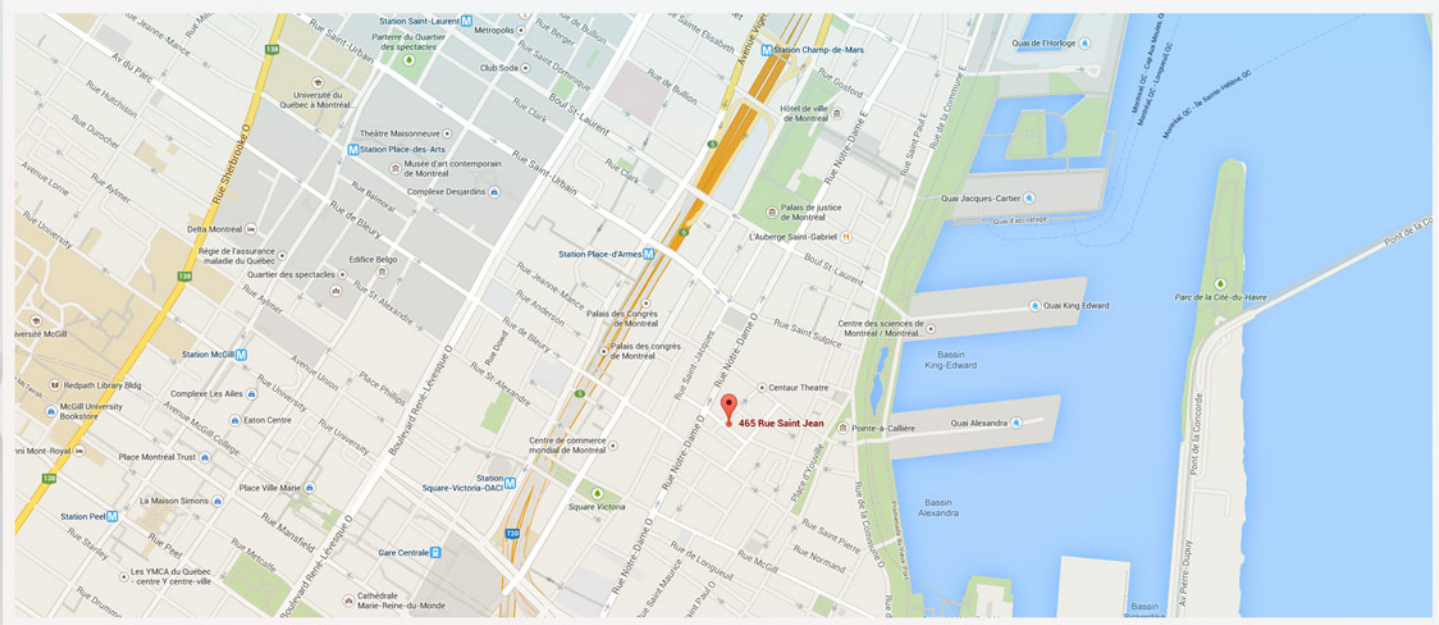
The AZUR Group's mission is to offer a competitive North American nearshoring business model by optimizing its Agile software development process to develop secure, high-performance, robust, and affordable software that grows with your business.

To further reduce our customers' software development costs, we use the XI-Factory tool to synchronize software development efforts from AZUR Group's experts, along cross-functional teams from our carefully-selected outsourcing partners.

AZUR Group's highly skilled IT business analysts, software engineers, developers, and database experts can perform a proof of concept with your team to validate FaaS efficiency for your organization.

Call us now and start modernizing your mission critical applications for the Cloud — and stay one step ahead of your competitors.

GET IN TOUCH



AZUR GROUP

T : 514 281 0061

465, St-Jean Street, suite 401
Old-Montreal, QC, Canada H2Y 2R6

contact@azurgroup.ca | www.azurgroup.ca | www.xifactory.com