

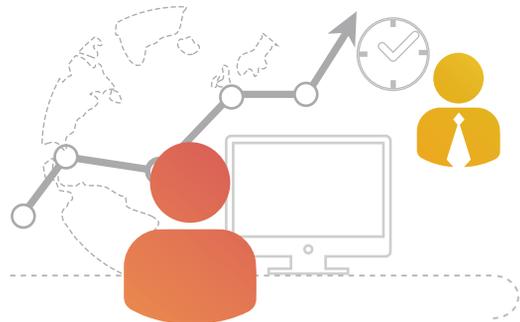
Co-Innovation with a
**Distributed
Software Factory**



Digital demands continuous value

In an exponentially changing digital landscape, software is the primary thread and driver in an ecosystem of connected products and services. In this ecosystem, where the business and customer have a direct engagement, often the software is delivered as an 'always on' service. In such an environment, the emphasis is not merely on delivering a set of features or functions, but also on supporting a continuously evolving user-experience. This new paradigm has a significant impact on how businesses approach software development.

The ability to build software in a traditional manner with a static process, is no longer sufficient. The need of the hour is to have a dynamic and agile process for rapid development and delivery of software enriched user experiences, or in other words, **capacity for continuous innovation** to meet the **digital demand for continuous value**.



“Software-driven businesses must support a continuously evolving user-experience”

Industrialization of software development

The often-heard grievance in the business world is “*why is it so difficult to industrialize the software development process*”? The obvious comparison is with the manufacturing industry’s approach (factory) for producing physical products. The key features of factory approach are **reuse (production infrastructure)**, **repeatability (standardized process)**, and **predictability (output and quality)**, which are the primary drivers of mass manufacturing, and delivered unprecedented economies of scale and quality. In this modern digital landscape, where software is the primary driver, the key question that must be addressed is, can you *industrialize software development*?

Technology disruption and continuously changing consumer preferences, makes it challenging to forecast what kind of software will address the market

demand. The best that any business can do is to build a robust organizational and process framework to withstand the disruption on multiple fronts, and is equipped with the right people to quickly develop and deploy a software product to meet the emerging market needs.



“Why is it so difficult to industrialize software development process?”

Software factory

In a connected, digital landscape where software is the primary driver, every business must think and function like a software company, and develop sophisticated capabilities in software-driven innovation. The demand for delivering continuous value and meet the evolving and changing consumer needs in a fast-paced technology world is driving the industrialization of software development, and the emergence of a **software factory**.

What is a software factory?

A **software factory** is a construct of people, processes, and tools, supported by automation and governance for continuous development and delivery of software applications. The continuous process of a factory setup is designed to ensure speed, predictability, and quality.

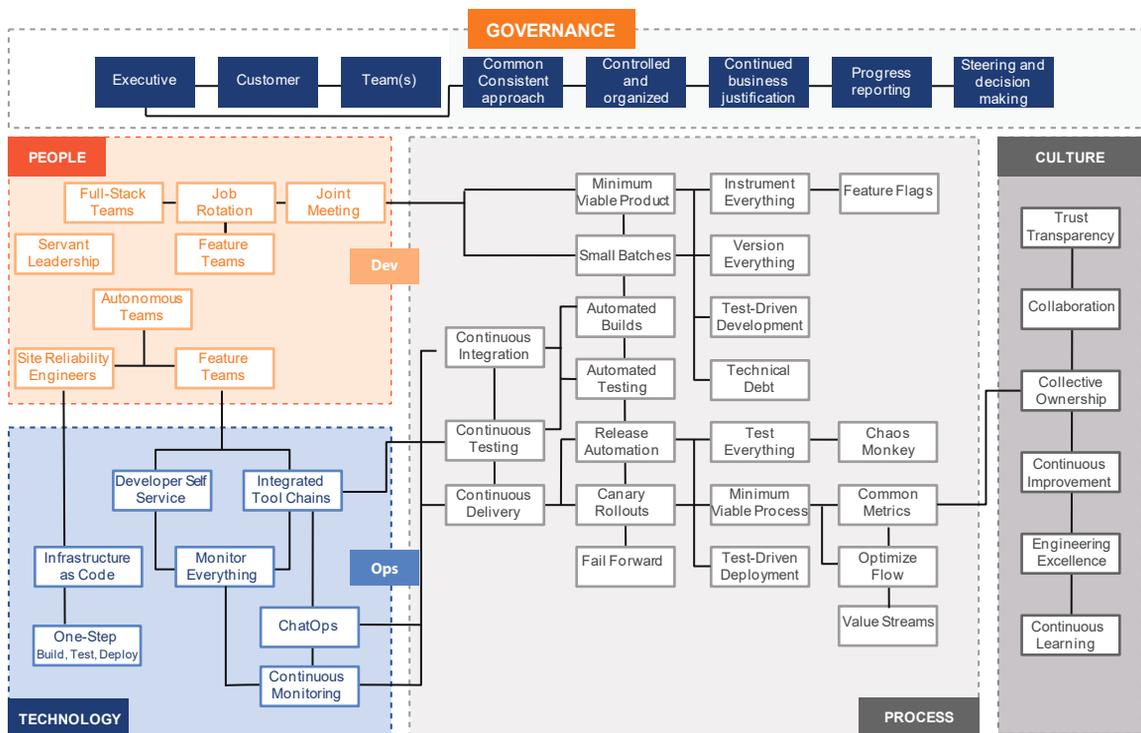


“Software factory is a construct of people, process and tools, supported by automation and governance”

Distributed software factory

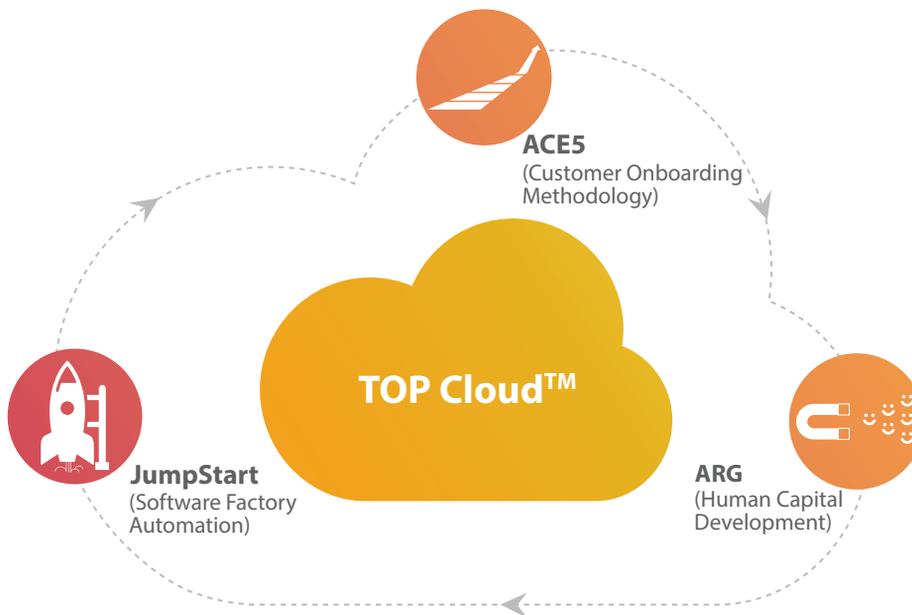
In this age of borderless innovation, it is but inevitable that software-driven businesses will seek the best talent from across the globe. The success of giants like Google, Apple, Facebook, Amazon and many others in leveraging global talent to realize their technology and innovation roadmaps, has proven beyond a doubt the efficacy of distributed development. Having a globally distributed development and service capability has become a key competitive differentiator for software driven businesses.

Distributed Development helps businesses to overcome a multitude of challenges and reap enormous benefits including, gaining access to highly skilled talent, operational flexibility, scalability, and cost optimization. Therefore, the need of the hour for software-driven businesses is to be equipped with a Distributed Software Factory, that is fully integrated and is capable of seamless delivery with contributions from geographically distributed teams.



TOP Cloud™

TOP Cloud™ is coMakelT's unique delivery platform that is used to setup a distributed software factory, tailored for the customer's specific needs, and consists of the following methodologies and services:



The various components of TOP Cloud™ are used in an integrated manner to operationalize the customer team, to setup and run a distributed software factory.

ACE5

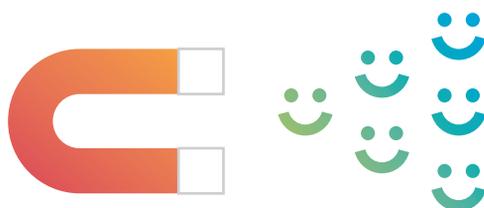
ACE5 is coMakeIT's pre-defined, onboarding methodology that takes the customer by hand, and guides them through every step of building a distributed software factory, in a structured manner. ACE5 guides the overall customer engagement from customizing the generic R&D organizational setup implementing the specific software factory model, setting up the team with appropriate process, to delivery, and encompasses all TOP Cloud™ components. ACE5 uses other TOP Cloud™ components such as ARG for recruiting and onboarding the team, and JumpStart to setup a development environment with the requisite process, tools, and automation. The different phases of ACE5, their objectives and deliverables are described below:



Phase of ACE5

| Phase | Objectives | Activities and Deliverables |
|------------|--|---|
| Align | Understand the customer's organization, technology, product, and process landscape, as well as goals, objectives, and timelines to build an outline for the engagement | Engagement Overview document outlining solution description, team setup, technology stack, proposed process, dev environment, and supporting workflow tools |
| | | High-level project plan for recruitment, onboarding, and ramp up with milestones and timelines |
| | | Selection of software factory model, Organizational blueprint with roles and responsibilities |
| | | Governance framework |
| Acquire | Using ARG, build the team and plan for structured onboarding | Customer team recruited and setup as per agreed upon plan |
| | | Plan for team onboarding and structured knowledge transfer and implementation initiated (to provide for phased onboarding) |
| | | Product backlog and prioritized work requirements |
| Assimilate | Onboard and integrate the team, and using JumpStart, establish distributed development environment | Completion of team onboarding and knowledge transfer as per plan |
| | | Distributed development environment established including version control, configuration management, build, integration, and release process |
| | | Common understanding achieved between the customer and distributed team on vision, goals, process, standards, roles & responsibilities, and expectations |
| | | Distributed Scrum process established for incremental development & delivery |
| Adjust | Produce initial results, optimize process to achieve predictable delivery and consistent quality | Optimized process and workflow |
| | | Metrics generated for assessment |
| | | Predictable quality and consistent delivery achieved |
| Accelerate | Establish distributed software factory, continuous improvement and continuous innovation | Distributed software factory established |
| | | Continuous improvement through governance framework, resulting in accelerated delivery with enhanced quality |
| | | Strategic and continuous innovation |

ARG



ARG (Attract-Retain-Grow) is coMakeIT's Human Capital Development (HCD) framework, and is used for recruitment, performance management, and career growth of all the people working for the customer teams. ARG is used by TOP Cloud™ component ACE5 for setting up and managing the customer team and the objectives and deliverables of its various components are described below:

| Component | Objectives | Activities and Deliverables |
|-----------|--|---|
| Attract | Recruit and setup a team as per the mutually agreed plan in ACE5 | HCD team initiates recruitment process in conjunction with customer stakeholders |
| | | Sourcing, evaluation, fitment, and shepherding |
| | | Employee on-boarding |
| Retain | Establish a progressive, result-oriented, objective culture and performance evaluation process | Cultural and organizational practices for keeping retention low, like frequent travel, and higher level of ownership and delegation |
| | | Objective goal setting, KRAs, performance evaluation, and recognition |
| | | Career development plan, identification of gaps, professional training, and mentoring |
| Grow | Aligning employee career goals and aspirations with organization and business goals | Competency development plan, tracking of core competencies of associates |
| | | Nurturing and training for chosen competency stream |
| | | Succession planning for key roles, and leadership training |

JumpStart



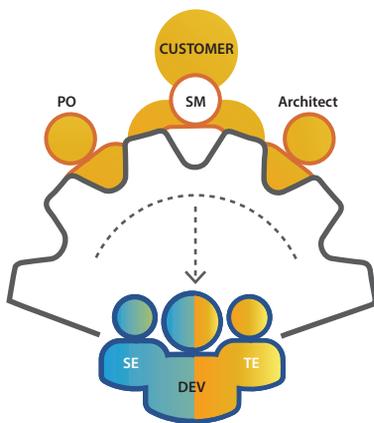
JumpStart is coMakeIT's way of enabling the customer's team with process, tools and automation. JumpStart is used by TOP Cloud™ component ACE5 to equip the customer team for continuous delivery with an automated framework for continuous integration, deployment, and testing (CI/CD/CT).

| Component | Objectives | Activities and Deliverables |
|-----------------------------------|--|---|
| Process | Create a development environment for continuous development and delivery | Workflow process established for creating a requirements backlog, validation, and prioritization |
| | | Workflow process established for architectural design review, and validation |
| | | Standards and process defined and established for development |
| | | Test strategy defined and implemented including process for code reviews, code coverage, test planning, test metrics, automated testing, and defect tracking |
| | | Process defined and established for code check-ins, source control, version control, daily builds, continuous integration, deployment and release |
| | | Scrum based PMO established for backlog management, sprint planning, and project tracking |
| Tools (for workflow & automation) | Support the process with the right tooling environment for all critical functions with an emphasis on automation | Select and deploy tools appropriate for the chosen technology stack including requirements management, design, IDE, CI/CD, test automation, source control, project management, communication, and knowledge repository |

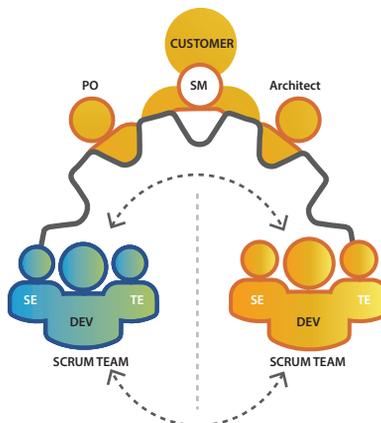
Distributed software factory models

Through our extensive insights and product development knowledge base, we help our customer's setup the following models of distributed software factories, with the flexibility and agility to evolve gradually from a basic model to a more mature model:

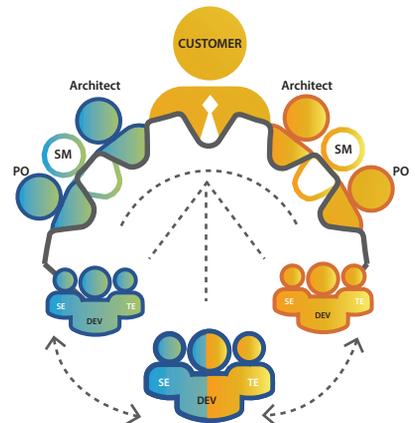
Mixed
scrum Teams



Autonomous scrum teams
working independently



Distributed scrum teams
working in conjunction

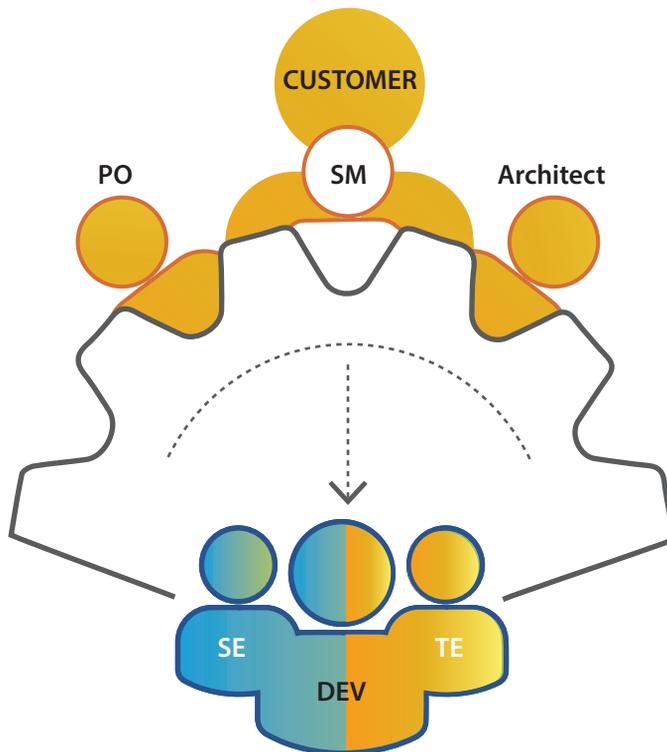


 Customer  coMakeIT

Mixed scrum team

This is a basic model of software factory setup that is relatively easier to setup and can start producing results in a short time. In this model, the process and workflow that is followed by the customer's native teams is replicated, and the distributed scrum team works as an extended team.

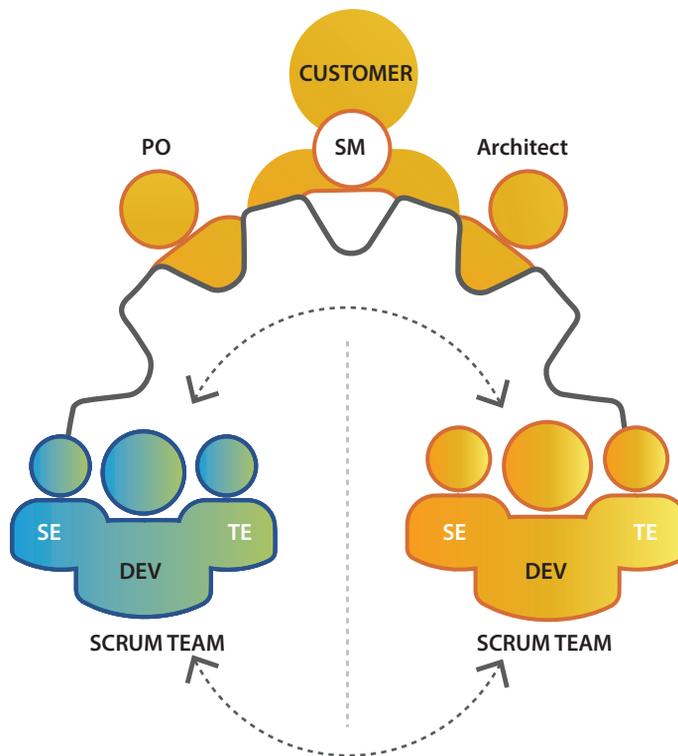
This model is best suited when the customer has the bandwidth to manage the distributed team, and wishes to retain control over all critical functions including architecture, product and project management.



Autonomous scrum teams working independently

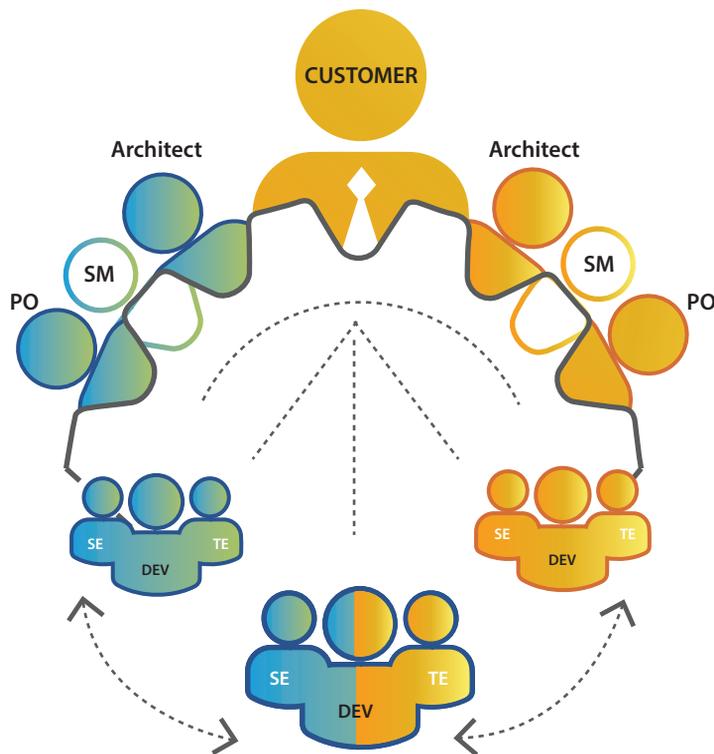
This is an intermediate software factory model, where a full-fledged distributed scrum team works in an autonomous fashion. The autonomous scrum team can either replicate the customer's native process or adopt a process better suited for its distributed environment.

A common team of enablers fulfilling critical functions such as architecture, product, and engineering management support both the native as well distributed scrum teams, which work independently. This model is best suited, when the customer has a clearly identifiable division of work between the teams, and both teams have adequate process and people maturity to deliver independently.



Distributed scrum teams working in conjunction

This is a mature model of software factory, with fully evolved processes and workflow to support distributed, yet collaborative development. In this model, the customer's native and distributed teams will be fully integrated in all aspects, and deliver seamlessly as a single entity. Geographic proximity becomes irrelevant in this model, with an integrated process and collaborative culture helping the teams work in tandem. Handling of critical roles and functions will be determined more by the availability of skilled talent and their level of expertise, and not by their location.



Software factory governance

Irrespective of the software factory model, a robust framework for program and process governance is critical for fulfilling the primary mandate of continuous delivery with quality. The governance framework should cover the strategic, operational, and tactical aspects of software development and delivery. In our experience, a 'RACI+F' framework (*Responsible, Accountable, Consulted, Informed, Facilitated*) can be of great help in charting people, roles, and responsibilities associated with all critical functions. It is important to remember that, in an agile environment with iterative delivery, governance is not static, but evolving continuously. A mature governance framework should have a mechanism in place to generate and capture metrics on all critical aspects, which should in turn lead to concrete and measurable improvements in the functioning of the software factory.

Metrics

Generating metrics on all key aspects is of vital importance for software factory governance. Any activity that can be tracked and measured must be tracked and measured. Metrics must be generated on all aspects of the software factory including people, process, and in a distributed environment, the partnership with service providers. Some key metrics that can be tracked are as follows:



Partnership – flexibility, scalability, risk-sharing and cost-optimization



Team – people, retention, technology skills



Process – velocity, predictability, quality

“A mature governance framework should have a mechanism in place to generate and capture metrics on all aspects of the software factory”



coMakeIT

continuous innovation

coMakeIT B.V.
Stationsplein 62, 3743 KM Baarn
The Netherlands T: +31 35-303 5630

coMakeIT Software Pvt. Ltd.
Plot No.564/A 39, Road No.92, Jubilee Hills
Hyderabad 500 033, India T: +91 40 4035 1000



coMakeIT is a ISO 9001:2015 and ISO
27001:2013 certified service provider
www.comakeit.com