# Turbo Charging IoT Projects with Agile Scrum Methodology

With the increasing adoption of the Internet of Things (IoT), companies are transforming daily business operations. Changing consumer purchasing behaviour has enormously impacted a product's shelf-life. Consumers today not only expect new features but also rapid product development cycles. Traditional project management is being replaced by a dynamic and future-ready approach. IoT products consist of both hardware and software offerings. While software service providers are familiar with large-scale, distributed, and time-boxed project management, many hardware teams still work in the traditional mode. Existing long-term development methodologies of hardware and software industries are being challenged by the requirements of a large distributed user-base and the need for rapid innovative features of IoT products.

In this whitepaper, we describe an Agile Scrum framework which can be used by an organization for IoT product development. It addresses the needs of both the end user and development teams. The focus is on customizing Agile Methodology that overcomes the IoT project challenges, with the rapid development of new product features, and continual software updates that influence a large consumer base and their purchasing behaviour.

# Tweaking Agile Scrum Methodology for IoT Projects

The IoT, a complex network of sensors, actuators combined with remotely managed software, and analytics is causing cross-industry disruption. With the advent of IoT-based products in the market, organizations worldwide are looking to leverage its power. The next few years will see an increase in the spend on the IoT by industries across the globe. According to the major global trend study on the Internet of Things, companies across the globe will be investing heavily on IoT initiatives

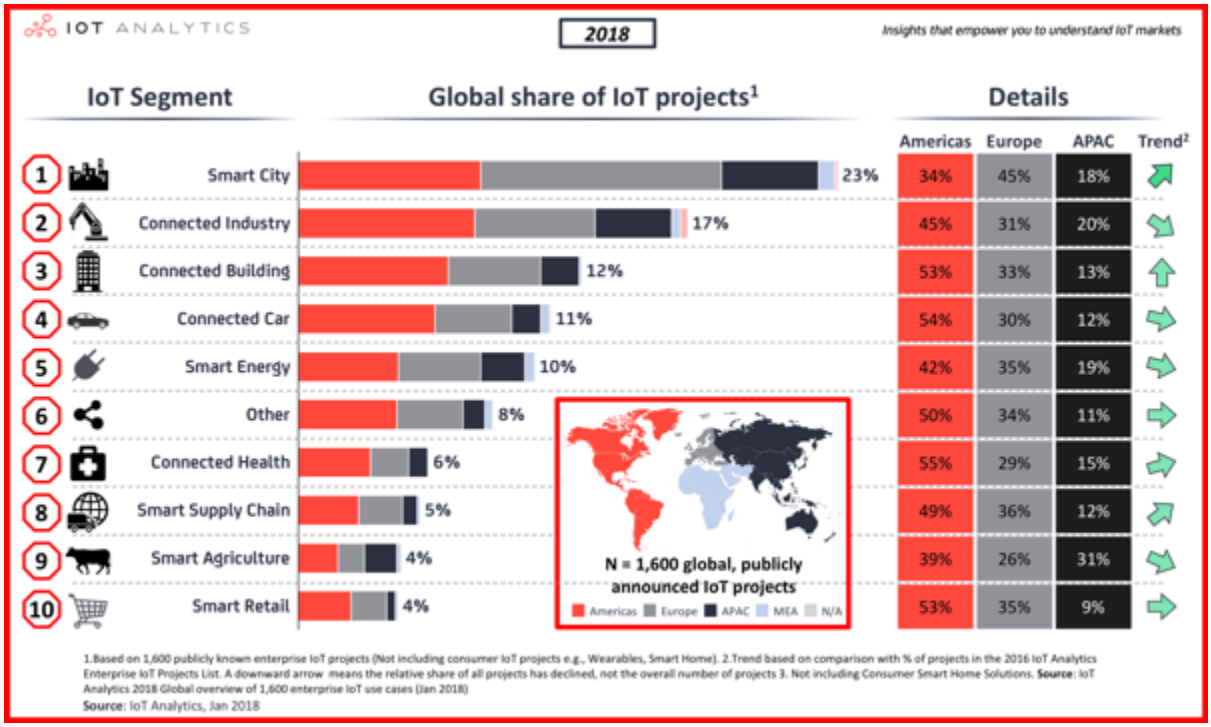


**Figure 1: Global share of IoT projects**

These IoT initiatives are innovative, with interconnected teams requiring multidisciplinary skills to offer faster deliverables, and cater to changing requirements in a single project. The IoT initiatives also throw open the possibility of leveraging dozens of new platform options, and hundreds of variations of hybrid IT and IoT integrations. As organizations continue to enrich products and related applications with increasingly rapid product development cycles, IT leaders will be forced to accommodate multi-disciplined technologies, hybrid integration platforms, and new IoT standards quickly. This dramatic shift creates new standards in the industry and requires a paradigm shift in application development.

While the Agile methodology has been widely referred to as a better fit for project traits like rapid development sprint cycles, distributed assets, and continual software updates, the conventional Agile approach may not be most suitable for IoT projects. This is because the use of nascent technology and evolving platforms implies that the teams should have cross-device expertise, and knowledge of firmware, networking, and analytics (including software) to deliver on objectives. Also, testing will require the use of simulation and non-simulation techniques, and longer integrated tests will be required than what is performed in conventional Agile projects.

In this paper, we focus on how a customized Agile-Scrum Methodology can be applied as a best-fit methodology for an IoT development project.

# Challenges Encountered

As organizations begin executing IoT projects they are likely to encounter some of the following challenges:

- **Demand for multi-technology and skills expertise:** IoT projects require the knowledge of various technologies and a combination of hardware and software skills. To understand the complex nature of an IoT project, see Figure 2.

The connected embedded device layer consists of smart devices widely dispersed among consumers. The devices connect with a gateway device or should include an embedded gateway module to connect with the platform layer. Teams working with these two layers need to be well versed with devices, firmware's SDK (Software Development Kits), and network protocols. The platform data aggregation layer is responsible for remotely managing devices and receiving the data from the gateway devices. The team needs to be skilled in device communication protocols and integration techniques to effectively manage data. The analytics platform provides reports and visualization of data. The team working on this layer requires Big Data, analytics, and machine learning skills. The applications could be developed as web or mobile applications, or be a part of existing enterprise applications.

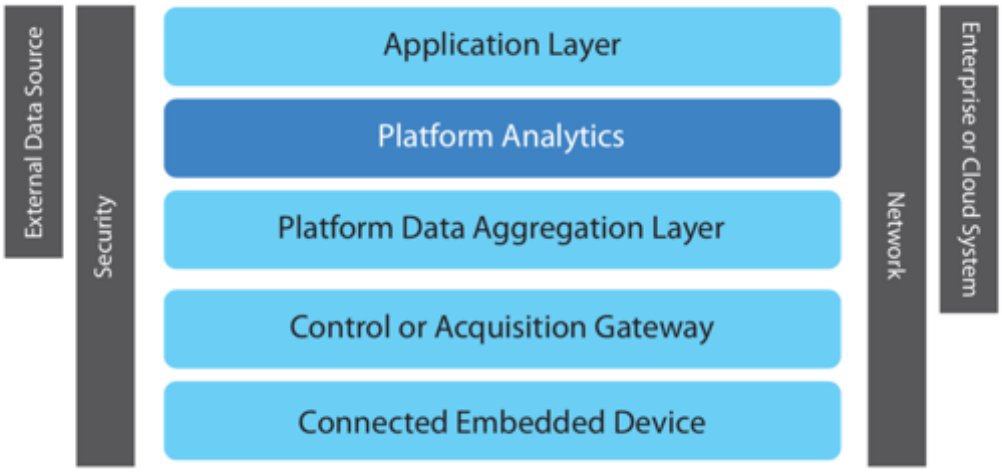


**Figure 2: Layers of IoT**

- **The need to accommodate evolving IoT standards across the product lifecycle:** IoT is a nascent technology undergoing rapid change and various standards are being adopted. The development teams need to be aware of all industry innovations and include appropriate standards to ensure that the product has an extended lifecycle.

- **Incremental design to solve hardware and software requirements throughout the product lifecycle:** IoT platforms operate on a low cost and a constrained bandwidth environment. The development methodology should ensure adequate design consideration that solves the functional and non-functional requirements. The build and release strategy also needs to keep these constraints in mind.

- **Suitable testing approach across heterogeneous engineering technologies and multiple cycles:** The testing needs should span heterogeneous technologies. The cost of creating an ideal testing environment for embedded devices is especially high. The chosen development methodology should accommodate the best alternative choice of integration testing with a heterogeneous choice of technologies.

- **Evolving security and privacy requirements throughout the product lifecycle:** As the IoT connects more devices together, it provides more decentralized entry points for malware, creating additional complexity and new security risks. The control of access to data may lead to compliance issues. New compliance frameworks may also evolve to address emerging IoT product requirements.

- **Continuous software updates and incremental deployments:** Products developed using IoT technologies need periodic updates that can even extend the life of the product. Hence, organizations developing IoT products need to think of testing and troubleshooting even obsolete products. An example is the Home Automation Platforms, as more and more smart home devices are available in the market, the home automation platforms need to support technologies and standards of new devices. Companies should offer upgrades to connect to newer device standards. Hence, the chosen development methodology should permit to extend the product lifecycle in the market

# Assessing Potential Issues in a Sample Logistics Project

Let us take a sample IoT project and understand the challenges encountered. Figure 3 provides an overview of a use case where a logistics company keeps track of temperature- and humidity-sensitive shipments with real-time tracking. This ensures that the shipped perishable items are delivered with the quality intact.

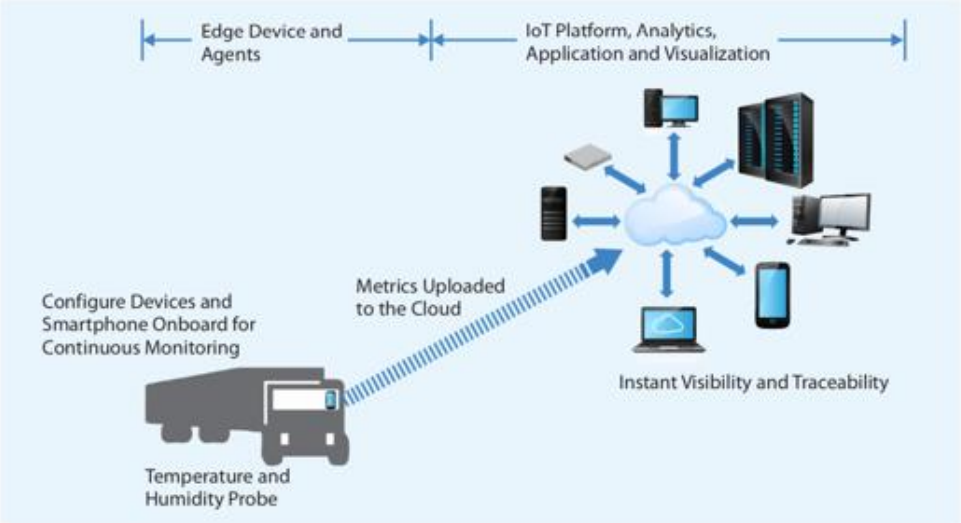


**Figure: 3 Key IoT Characteristics of a Sample Project**

In such an engagement, the project team would have to consider the following challenges for each layer of the IoT framework encountered:

| IoT Layer | Key Challenges |
|---|---|
| Connected Embedded Device | • Need to identify appropriate sensors and finalize device connectivity protocol such as Zigbee, Bluetooth, and so on<br>• The device should be able to operate and connect in harsh conditions (high pressure, humidity, and rough handling)<br>• The team may need to experiment with these protocols and devices, and needs to consider evolving IoT standards |
| Control or Acquisition Gateway | • The team needs to identify low-cost connectivity options and ensure that the device is able to connect to the platform even in remote places<br>• The device needs to be able to receive software updates, authenticate and certify it, and consider evolving security and privacy requirements |
| Platform Data Aggregation Layer | • The platform should support the protocol of the gateway agent<br>• The platform should cater to changing technology and trends<br>• The team should have cross-skill knowledge with the ability to easily develop the evolving changes. The team should also automate processes through which continuous software updates and incremental deployments are shipped |
| Platform Analytics | • The platform should support analytics, reporting, visualization, and machine learning<br>• The team should be able to understand the new type of intelligence that is discovered from hardware and software information |
| Application Layer | • The team should be able to design intuitive applications that suit devices |

# Introducing the Agile Scrum Methodology for IoT Projects

The Agile Scrum methodology is suitable for projects requiring rapid development and incremental delivery. However, IoT projects require hardware features to be developed aligned to software updates. This requires the customization of Agile methodology. The changes required are in terms of:

- How the teams are formed
- The Sprint initiation, and setting the architecture and development standards
- How the teams collaborate
- Extended integrated testing for device, gateway, and software teams

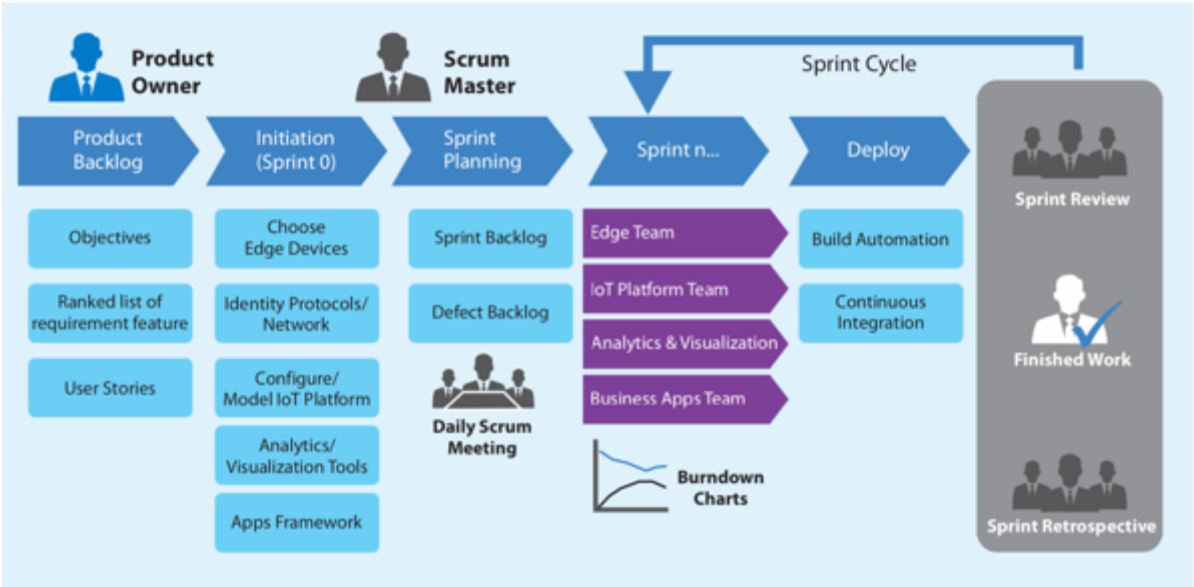


**Figure: 4 Key Agile Characteristics of a Sample Project**

# Team Formation: Roles Required for IoT Projects

IoT projects are multidisciplinary, driven by leading-edge technologies, and entail high risks. Agile can help improve risk prediction with proper planning. The typical roles required in an IoT project, driven by Agile Scrum Methodology, are:

- **Central Project Team:** A team that governs overall project management

- **Product Owner:** A member of the Agile team responsible for defining and prioritizing the team backlog. He does so, while maintaining the conceptual and technical integrity of the features initially planned.

- **Scrum Master:** An executive who makes sure that the Scrum team lives by the values and the practices of Scrum.

- **Scrum (Sprint) Team:** The development team responsible for delivering incremental products at the end of each sprint.

For the IoT project to be successful, it is essential that the IoT project team comprise of the Edge Team that enables devices to connect securely to the centralized IoT Platform. The IoT Platform Team enables device abstraction, management, and operations through a centralized console. This team creates a data model that supports the seamless connectivity of heterogeneous devices and smooth flow of its events. The Analytics and Visualization Team suggests complex event processing algorithm and machine learning library to be implemented in the analytics platform. Finally, the Business App Team enables business process management strategy.

# Sprint Initiation: Identifying Architecture and Standards

The Sprint initiation phase, which is not time-boxed like the regular sprint cycle, is where the project team develops the business case and vision for each release, assembles the team, and identifies the role of each member. The initial product backlog is reviewed along with project management, shared resources, development and operations (DevOps), user experience (UX), system architecture, and release management teams, to ensure that activities are sequenced to produce the right project schedule. Sprint initiation lays the foundation for the IoT project, with a few additional steps like:

- Edge device identification to select devices suited for the business requirement
- Protocol or network identification to decide on the protocol and network
- IoT platform selection and configuration to select the requisite platform and configuration
- Data analytics visualization and the selection of application development frameworks

These architectural choices need to keep in mind the continuous software updates over the life span of the devices, and the key challenges identified earlier. Also, these choices will need to be made in consultation with the multi-skilled team.

# Key Stages of Agile Scrum Methodology

The Agile Scrum Methodology consists of the following phases:

▪ **Product backlog:** This is the ranked list of the requirement feature—a prioritized list of work for the development team that is derived from the objective, roadmap, and its requirement of the product. The most important items are presented at the top of the product backlog, so that the team knows what to deliver first. The size of these backlog items is usually referred to in user stories.

▪ **Sprint Planning:** This is an activity in Agile Methodology at the start of every sprint, for defining and prioritizing the team backlog. This is performed (as depicted in Figure 4) while

Sprint Planning incorporates:

• **Sprint backlog:** A list of tasks identified by the Scrum team to be completed during the Scrum sprint

• **Defect backlog:** A list of defects identified in the system during the testing phase of previous sprints. Some or all these defects are identified along with the task item to be completed during the Scrum sprint.

For instance, the sprint backlog of the cold chain example (in Figure 3) may have had an average of 19 user stories in each sprint, and one or two user stories of defects are prioritized with the defect-fixing scrum team.

• **Scrum Daily Meeting:** Each Scrum team attends the daily Scrum meeting, in which three major questions are asked. It is important that each team collaborate with the other with a common objective and prioritize the project needs accordingly :

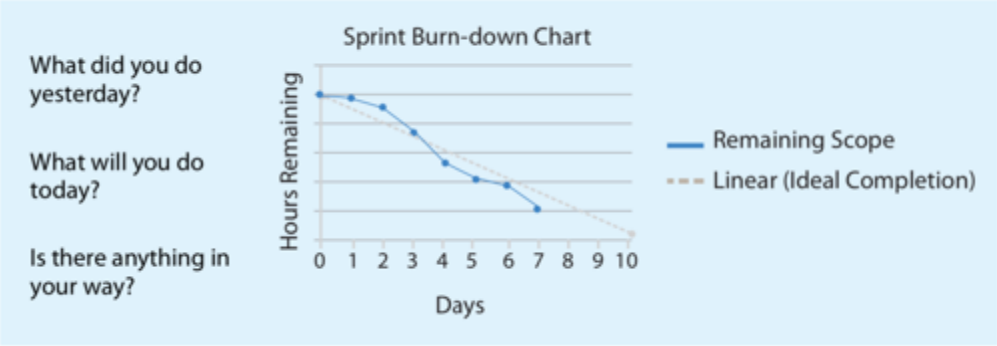


**Figure 5. The Sprint Burn-down Chart**

For instance, the project example (in Figure 3) included 15 minutes of daily Scrum meetings with the Scrum masters of each Scrum team.

▪ **Sprint Review Phase:** In this segment, project team members identify functionality that was not delivered or was not delivered as expected, and request that functionality be placed in the product backlog for prioritization. This exercise is typically carried out after integration testing, and while having a demo with the product owner and other stakeholders. The team then discusses potential re-arrangements of the product backlog with the stakeholders, based on the feedback in the sprint. In the IoT project, a single team cannot perform this action. This should be a joint review with multiple teams, as described at the start of this section.

▪ **Sprint Retrospective Phase:** In this stage, at the end of every sprint, stakeholders try to ensure that the project team is improving the way it works and devising solutions for emerging critical problems. This helps the team meet IoT project objectives.

# Extended Integration Testing

Integration testing in an IoT project involves testing hardware and software and transmitting massive amounts of intelligent information. With millions of sensors and devices working together with software in an IoT system, mere functional validation is not sufficient.

For instance, in the cold-chain example quoted earlier in the paper, sensors and back-end systems work together to ensure that the perishable product is delivered with a high degree of quality. As a real-time scenario is complex, integration testing demands a simulation process in the test environment. While hardware is well tested by device makers, testing the application intelligence along with devices is a complex task. A traditional approach for testing includes unit, integration, and system testing. But testing IoT platforms should include noise detection in the system, discovering boundaries of desired performance, and anticipating new behaviour and/or process changes based on the new ecosystem. This will involve creation of extended simulated lab conditions and user experience and behaviour testing.

# Conclusion:

IoT technologies are maturing fast and the markets are showing signs of massive adoption of connected devices and objects. Companies need to adopt the IoT to lead in the new competitive environment of smart, connected products. In most planned development methodologies of the past, we have seen engineers work in silos on a business function. The shipment of business critical functions would be released towards the end of the development cycle. In IoT projects, siloed performance is neither encouraged nor feasible, due to the multidisciplinary and interconnected nature of engagements. Furthermore, the cost of a product varies with the length of the product lifecycle. It increases exponentially for defects found in later stages. Figure 6 shows the relation of growth in a longer development lifecycle.

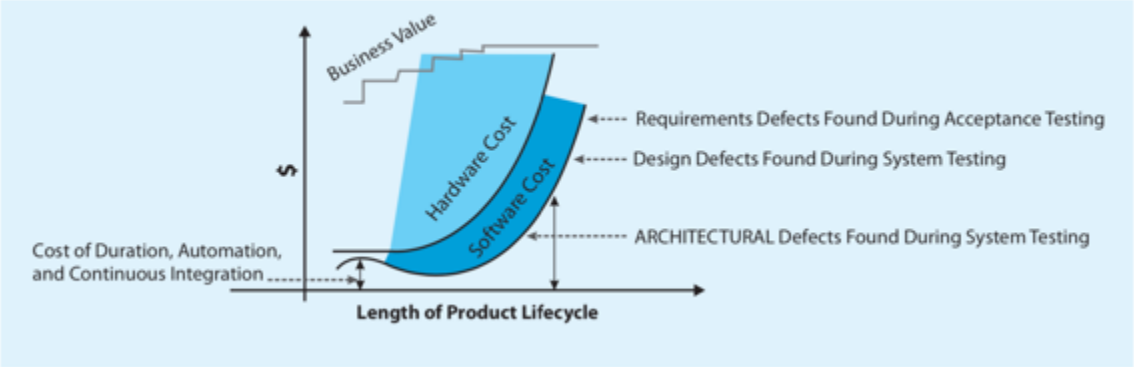


**Figure 6. Relation of Cost and Length of Product Lifecycle**