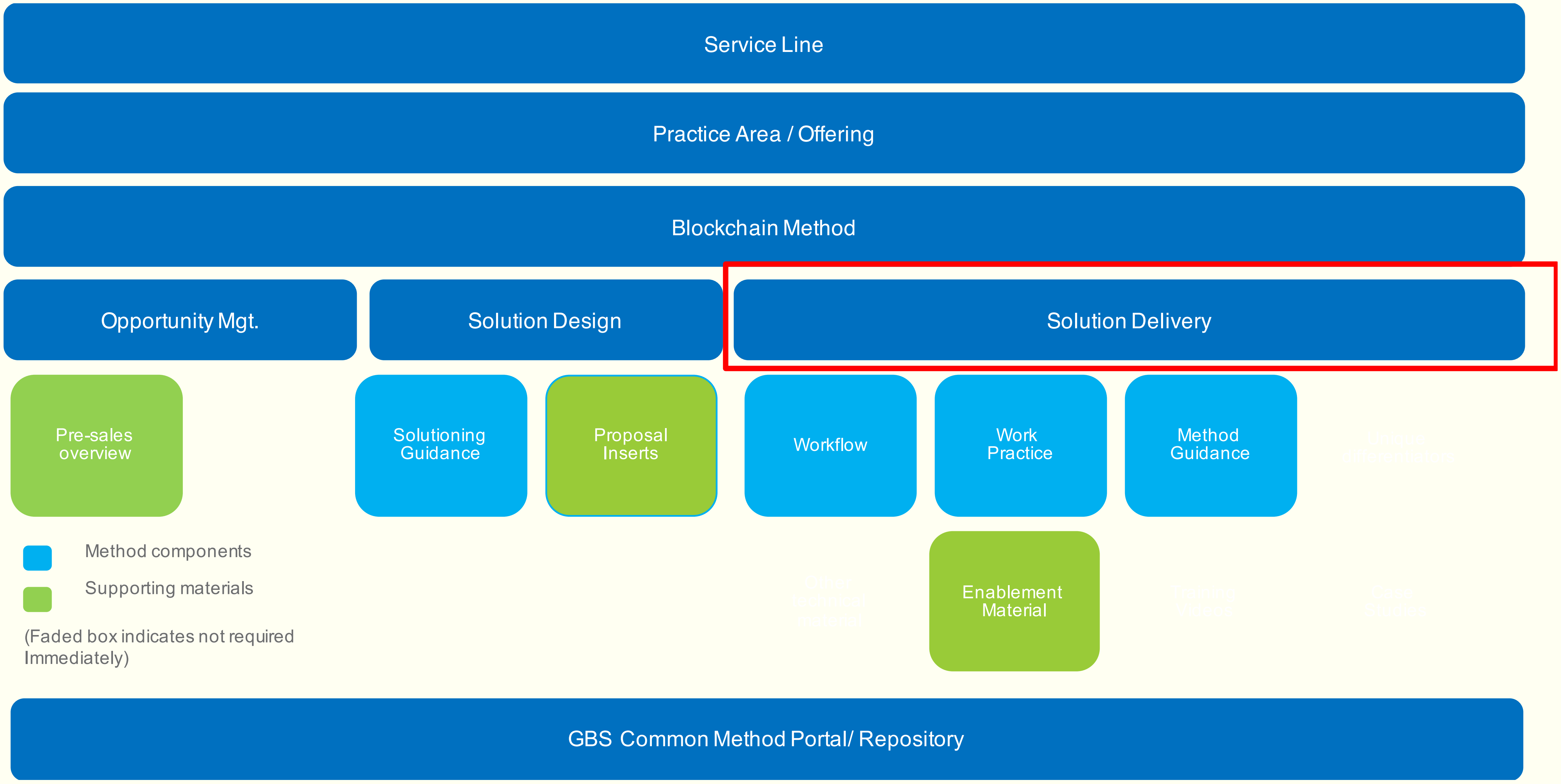
WORK PRACTICE

Blockchain Innovation

GBS Method

WORK PRACTICE

IBM Blockchain Innovation -Solution Delivery



Service Line

Practice Area / Offering

Blockchain Method

Opportunity Mgt.

Solution Design

Solution Delivery

Pre-sales overview

Solutioning Guidance

Proposal Inserts

Workflow

Work Practice

Method Guidance

Unique differentiators

Other technical material

Enablement Material

Training Videos

Case Studies

🟦 Method components

🟩 Supporting materials

(Faded box indicates not required Immediately)

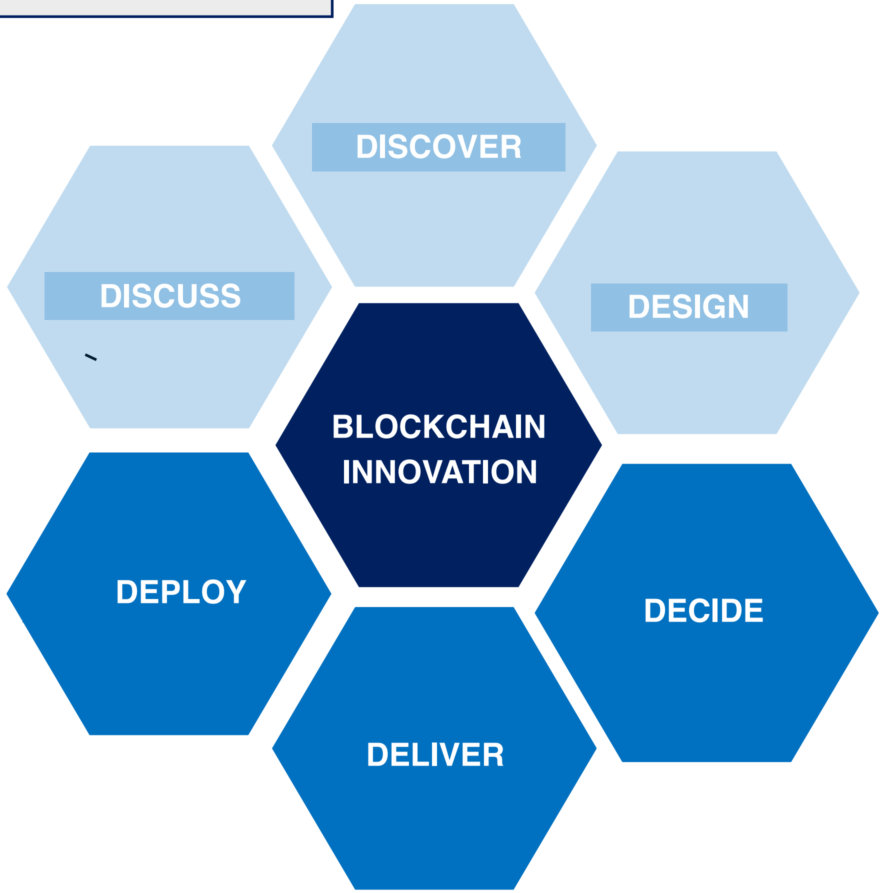
GBS Common Method Portal/ Repository

WORK PRACTICE

IBM Blockchain Innovation

IBM Blockchain Innovation provides a practitioner friendly, tool based, light weight approach to deliver applications using Agile principles enabled by automation and DevOps.

IBM Blockchain Innovation		
Opportunity Mgmt	Solution Design	Solution Delivery



- Plan and conduct user acceptance testing (UAT) for the Blockchain solution.
- Deploy solution after each increment using DevOps Tools.

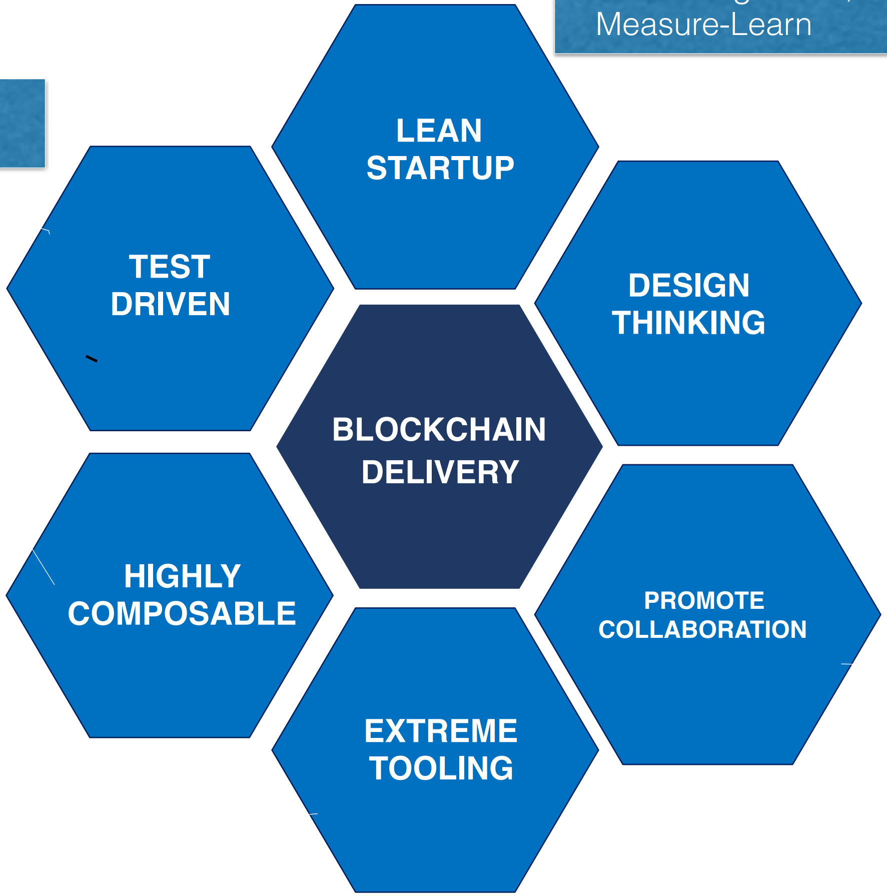
- Design, develop, integrate and test the Blockchain solution.
- Adopt established delivery processes, tools and guidelines

- Collaborate with client to finalize strategy, processes and technology for delivery
- Review Backlog & Sprint plans
- Validate the infrastructure, connectivity and tools

Methods

Design Principles for Blockchain Delivery

IBM’s method for Blockchain provides a practitioner friendly, tool based, light weight approach to deliver applications using Agile principles enabled by automation and DevOps.


LEAN STARTUP

TEST DRIVEN

DESIGN THINKING

BLOCKCHAIN DELIVERY

HIGHLY COMPOSABLE

PROMOTE COLLABORATION

EXTREME TOOLING

- Less Documentation, Digital Media & Cognitive analysis, Quick turn around, Eliminating waste, Build-Measure-Learn

- Test cases centric development

- A highly composable method provides components that can be selected and assembled in various combinations to satisfy specific project requirements.

- User centric focus, Cross discipline collaboration, Quick iteration, Stakeholder interviews, Empathy mapping, Minimum Viable Product, User story approach – All results in Delivering Great Experience

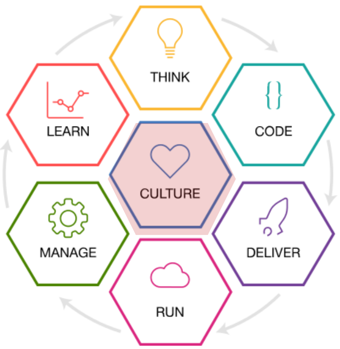
- Promote collaboration and communication for collocated as well as distributed teams. Promote usage of digital collaboration platforms and reduce communication gaps , enhance trust within and across teams.

- Promote, Standardize, Optimize Tool usage in each and every phase of the delivery lifecycle to bring in

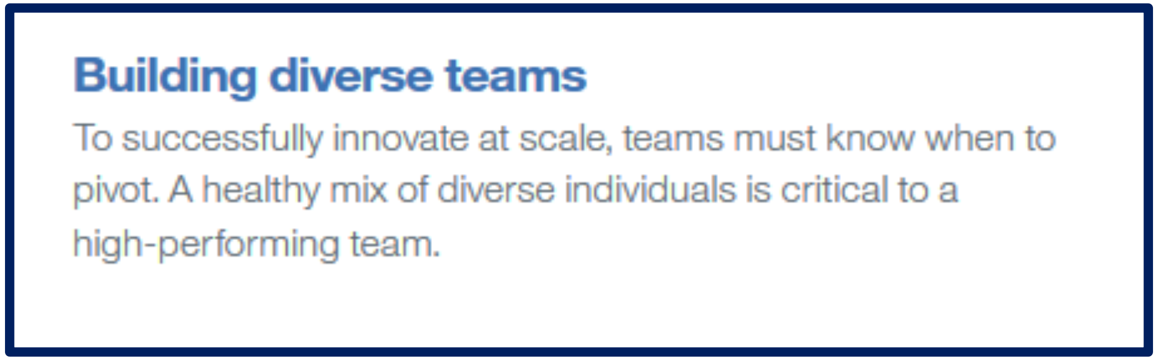
Decide – Work with all stakeholders for successful commencement of delivery

	Project On-boarding	Project Planning and Governance	Design Guidelines	Architecture and Technical Design	DevOps Delivery Procedure
ACTIVITIES	▪ Onboard PM and key resources ▪ Register BMS Account ID & Generate work items ▪ Conduct Project Kick-off ▪ Review SoW, Scope Solution & IBM commitments	▪ Conduct Project planning workshop ▪ Identify User stories for initial three sprints ▪ Define Roles & Responsibilities ▪ Define Test strategy ▪ Review & confirm plans with key stakeholders ▪ Initiate environment setup	▪ Define technical stack for Chain code and UI ▪ Finalize the Devops strategy and tools ▪ Define Security **parameters** and Data Privacy ▪ Define scope and requirements for Smart Contract	▪ Execute technical workshop ▪ Review existing IT landscape & standards ▪ Develop architectural decisions ▪ Develop end-to-end Blockchain architecture ▪ Identify & agree NFRs ▪ Define integration and security approach ▪ Define technical dependencies ▪ Leverage and optimize code	▪ Execute DevOps workshop ▪ Define key delivery processes ▪ Install & configure tools ▪ Establish client connectivity ▪ Initiate DevOps platform ▪ Configure platform access ▪ Identify dashboard requirements ▪ Define configuration management tools & procedure
DELIVERABLES	▪ Project Charter ▪ Kickoff and Project charter.	▪ Governance & Escalation Matrix, ▪ Project Organization Chart ▪ Weekly Project Reports ▪ Prioritized list of backlog stories ▪ Project Sprint Plan & Schedule ▪ RACI Matrix ▪ Test Strategy	▪ Data Security and Privacy assessment	▪ Architecture Decisions document ▪ Architecture Overview diagram and High level design document.	▪ Updated Project Handbook ▪ DevOps Setup
TOOLS, ASSETS	▪ PMP Tool and MS Project, JIRA, MS Excel	▪ JIRA, MS Excel, MS PPT	▪ Keynote / Power Point/Visio	▪ MS Visio, Drawing tools, MS Word/Powerpoint	

Decide – Core practices that will help in building trust and alignment

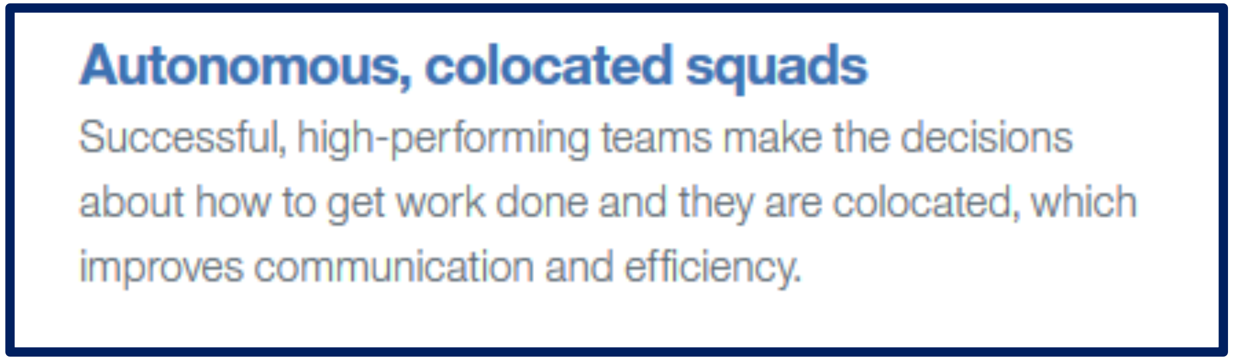


Culture



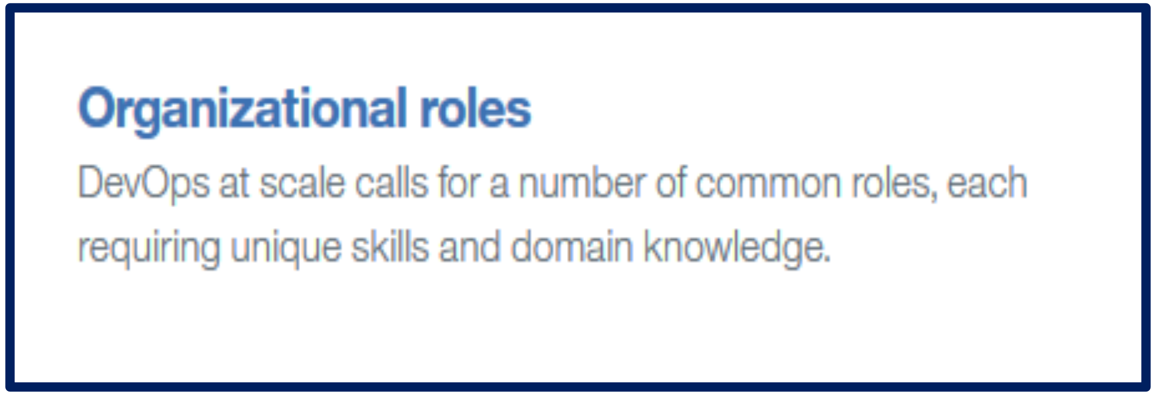
Building diverse teams

To successfully innovate at scale, teams must know when to pivot. A healthy mix of diverse individuals is critical to a high-performing team.



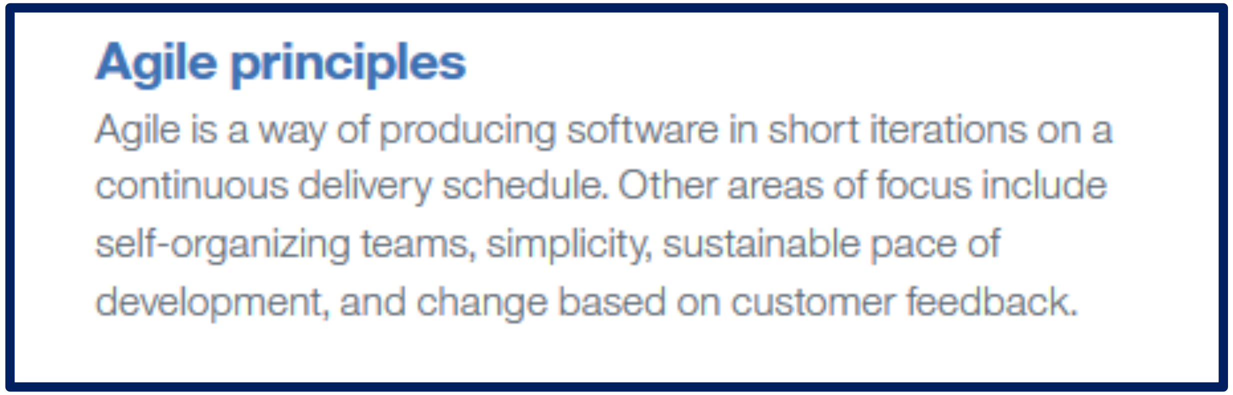
Autonomous, colocated squads

Successful, high-performing teams make the decisions about how to get work done and they are colocated, which improves communication and efficiency.



Organizational roles

DevOps at scale calls for a number of common roles, each requiring unique skills and domain knowledge.



Agile principles

Agile is a way of producing software in short iterations on a continuous delivery schedule. Other areas of focus include self-organizing teams, simplicity, sustainable pace of development, and change based on customer feedback.

Link to IBM Cloud Garage Method site for all practices

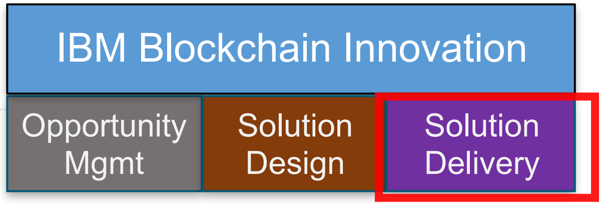
WORK PRACTICE

Deliver – Design, Develop, Integrate and Test Solution in iterative manner

	Due Diligence	Sprint Planning and Review	UX Design	Technical Design	Build, Integrate and Test
ACTIVITIES	▪ Backend / Infra readiness assessment ▪ DevOps impacts and assessments ▪ Start engaging other teams / Business Units as required ▪ Review Project plan with relevant stakeholders	▪ Generate and review project backlog and prioritize ▪ Capture user story description (business logic, assumptions and risks) ▪ Estimate Story points and Sprint Velocity to plan the backlog ▪ Define test plan including test data requirement ▪ Develop test cases ▪ Identify & Communicate client dependencies	▪ Create navigation flows for the scope of work in the sprint ▪ Develop Wireframes and visual design for the approved wireframes ▪ Conduct user testing	▪ Identify security and performance parameters ▪ Identify UI components and libraries ▪ Develop detail technical designs ▪ Create API definitions & validate API readiness ▪ Leverage assets & accelerators ▪ Review MDM, Hosting, Middleware impacts	▪ Develop user interface, backend logic & integration ▪ Perform unit testing and deploy solution across environments ▪ Perform static code analysis ▪ Collaborate effectively for defect tracking ▪ Create test cases & test data ▪ Perform System Integration Test ▪ Perform security & performance testing ▪ UAT preparedness and readiness
DELIVERABLES	▪ Finalized backlog with the customization and enhancements ▪ Prioritized requirements and complete Due Diligence Checklist	▪ Requirements Document with Acceptance Criteria. ▪ Sprint Plan, Test Plan, ▪ Client Dependencies	▪ UX Design Artifacts (personas, journeys, wireframes)	▪ Architecture Decisions document ▪ Architecture Overview diagram ▪ High Level Design Document	▪ Test Results ▪ Production ready application in various environments
TOOLS, ASSETS	▪ Jira	▪ Jira	▪ MS Power Point ▪ Photoshop	▪ MS word, Swagger, SOAP UI	▪ Fabric Composer, Jenkins, JUnit, SonarCube, Cucumber, Git, Selenium

WORK PRACTICE

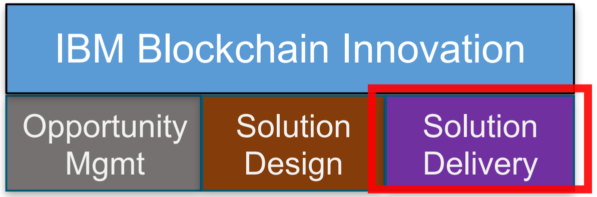
Deploy – Launch Blockchain Solution



	User Acceptance	Deploy	Launch	Hypercare
ACTIVITIES	▪ Validate UAT test plan ▪ Create UAT test data ▪ Perform / Support User Acceptance Testing ▪ Record test results ▪ Fix defects (UI / integration & functional) ▪ Receive UAT Sign-off ▪ Evaluate Deployment preparedness and readiness	▪ Create production build ▪ Data Migration activity (if applicable) ▪ Install adapters and integrator in the production environment ▪ Complete production verification testing (client-driven) - PVT ▪ Receive PVT sign-off ▪ Evaluate Application launch preparedness & readiness	▪ Create training documentation for end-users ▪ Support any client requested training ▪ Validate support readiness	▪ Full development support ▪ Defect Monitoring & Fixing ▪ Ensure system availability ▪ User level training ▪ Develop handover doc preparation ▪ Build to RUN transition initiation
DELIVERABLES	▪ UAT Test Plan ▪ Test Results ▪ Solution Packaging & Deployment Artifacts	▪ Deploy documentation	▪ Training Document (User Manual) ▪ Project closure report	▪ Updated Training Manual
TOOLS, ASSETS	▪ JIRA for defect management ▪ DevOps Tools & Processes, Deployment Preparedness Checklist	▪ Jenkins	▪ MS Word,	

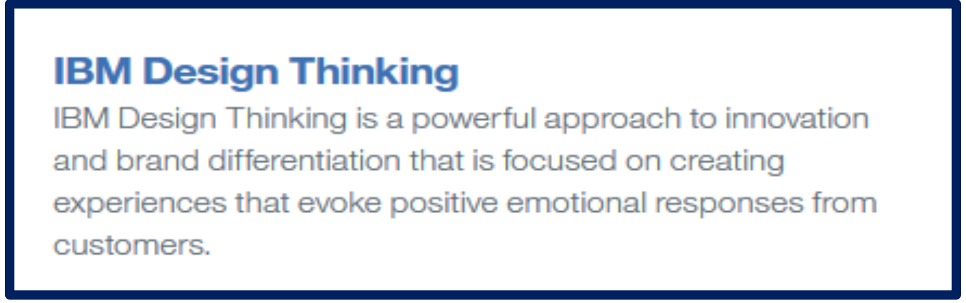
Deliver – Key practices to create and continuously integrate high quality code (contd.)

Think



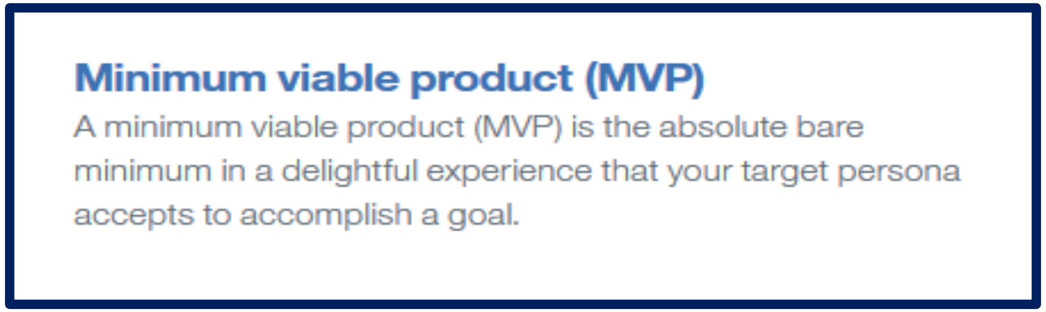
IBM Blockchain Innovation

Opportunity Mgmt | Solution Design | Solution Delivery



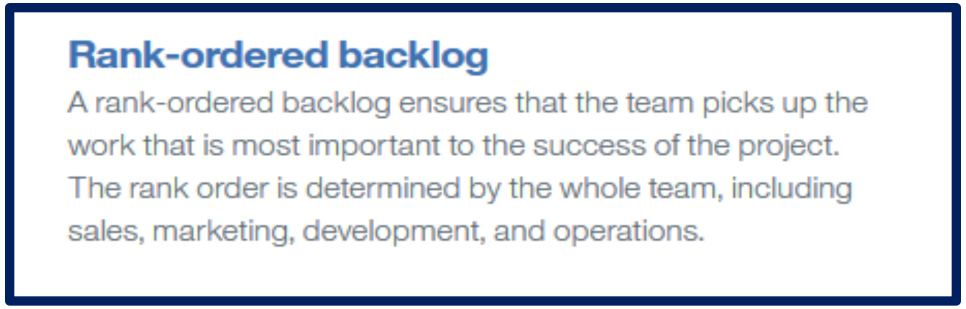
IBM Design Thinking

IBM Design Thinking is a powerful approach to innovation and brand differentiation that is focused on creating experiences that evoke positive emotional responses from customers.



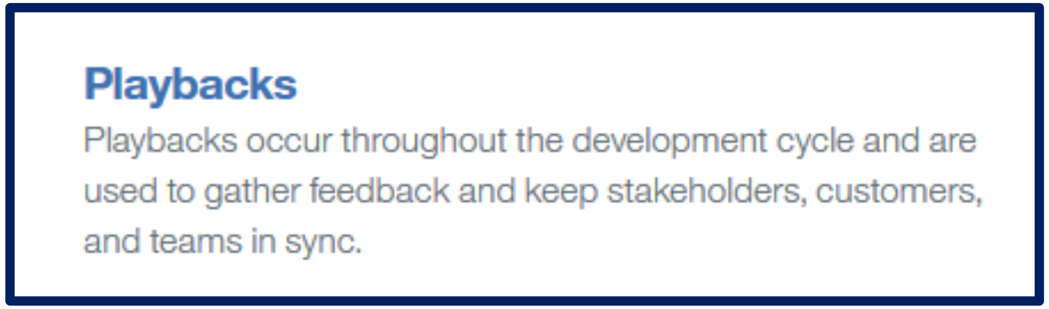
Minimum viable product (MVP)

A minimum viable product (MVP) is the absolute bare minimum in a delightful experience that your target persona accepts to accomplish a goal.



Rank-ordered backlog

A rank-ordered backlog ensures that the team picks up the work that is most important to the success of the project. The rank order is determined by the whole team, including sales, marketing, development, and operations.



Playbacks

Playbacks occur throughout the development cycle and are used to gather feedback and keep stakeholders, customers, and teams in sync.

Evolutionary Architecture

* Perform architecture and design work "just in time."
* Prioritize on mitigating technical risk over future value creation.
* Document key decisions and outstanding issues.

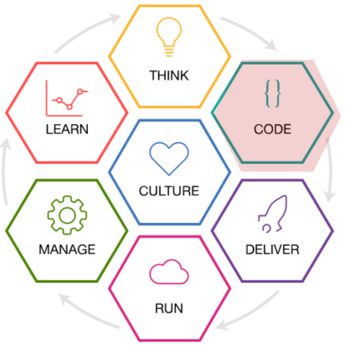
Estimation

* Ensure that an estimate is based on the template that the organization mandated.
* Ensure the estimate uses the best information available.
* Revisit estimation at the beginning of each sprint.

Link to IBM Cloud Garage Method site for all practices

WORK PRACTICE

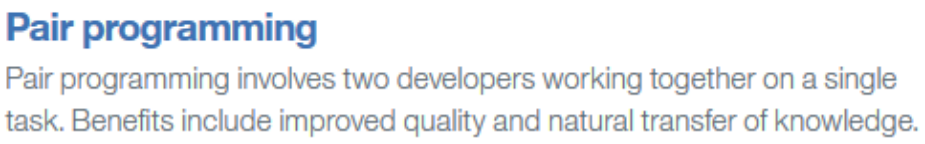
Deliver – Key practices to create and continuously integrate high quality code (contd.)



Code

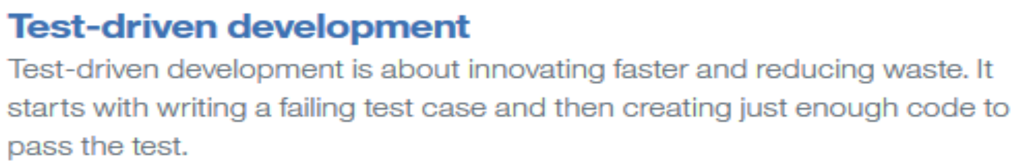
IBM Blockchain Innovation

Opportunity Mgmt | Solution Design | Solution Delivery



Pair programming

Pair programming involves two developers working together on a single task. Benefits include improved quality and natural transfer of knowledge.

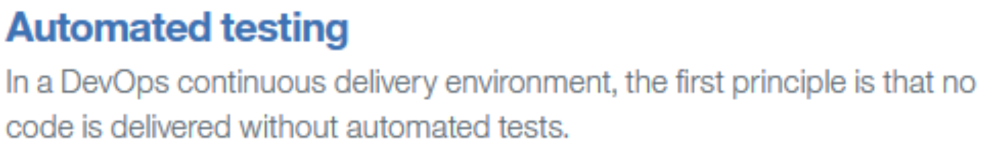


Test-driven development

Test-driven development is about innovating faster and reducing waste. It starts with writing a failing test case and then creating just enough code to pass the test.

Continuous integration

Early error detection, early system integration, and improved collaboration are the hallmarks of a high-quality product development process. One outcome is greater predictability in delivery schedules.



Automated testing

In a DevOps continuous delivery environment, the first principle is that no code is delivered without automated tests.

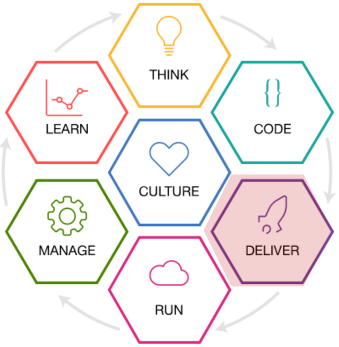
Build verification testing and integration verification testing

* Include critical functional tests that are related to the changes that were made in the application

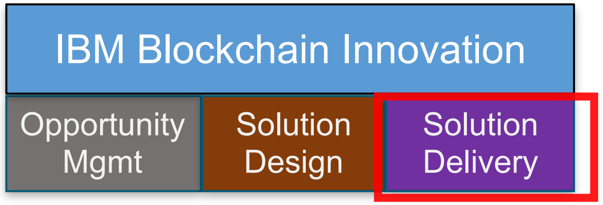
* Detect any integration issues and conflicts early in the cycle.

Link to - IBM Cloud Garage Method for all practices

Deliver – Key practices to create and continuously integrate high quality code



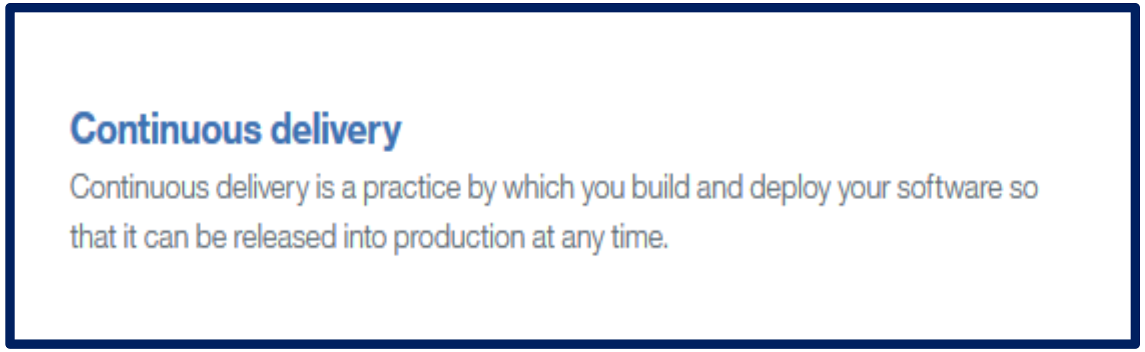
Deliver


IBM Blockchain Innovation

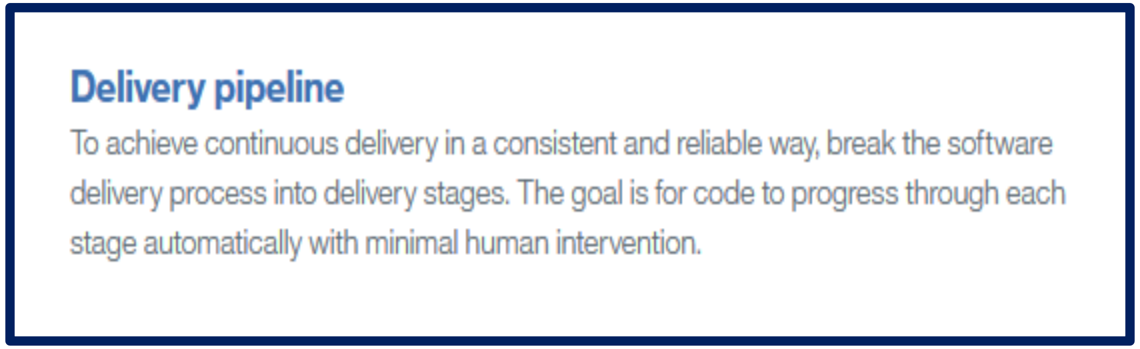
Opportunity Mgmt

Solution Design

Solution Delivery


Continuous delivery

Continuous delivery is a practice by which you build and deploy your software so that it can be released into production at any time.


Delivery pipeline

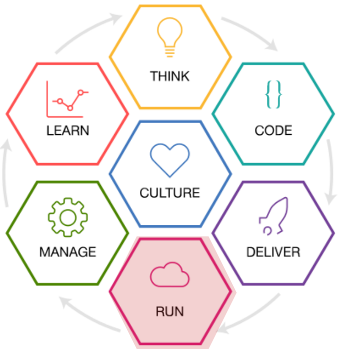
To achieve continuous delivery in a consistent and reliable way, break the software delivery process into delivery stages. The goal is for code to progress through each stage automatically with minimal human intervention.

Tools

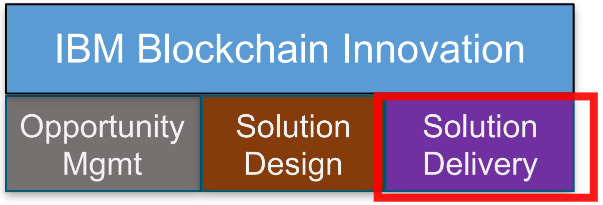
- Jenkins
- IBM Urban Code
- IBM Bluemix Continuous Delivery
- Artifactory
- Delivery Pipeline

Link to IBM Cloud Garage Method site for all practices

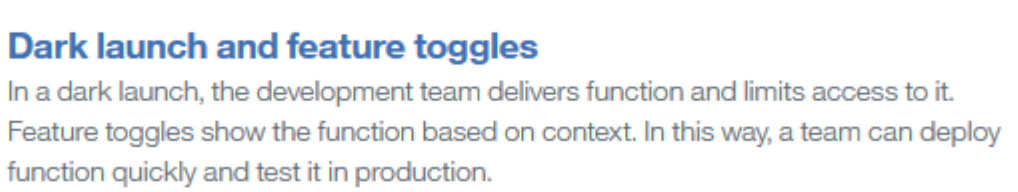
Deploy – Key Practices to get Blockchain Solutions to Production



Run

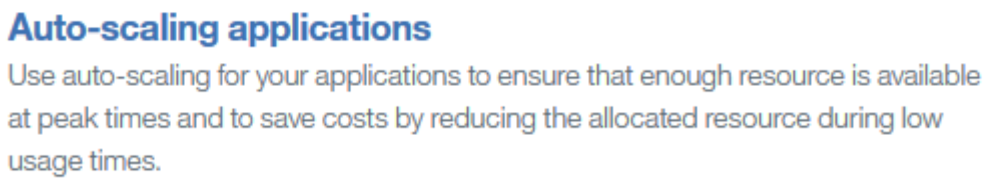

IBM Blockchain Innovation

Opportunity Mgmt | Solution Design | Solution Delivery



Dark launch and feature toggles

In a dark launch, the development team delivers function and limits access to it. Feature toggles show the function based on context. In this way, a team can deploy function quickly and test it in production.



Auto-scaling applications

Use auto-scaling for your applications to ensure that enough resource is available at peak times and to save costs by reducing the allocated resource during low usage times.

Monitoring on IBM Bluemix

Tools

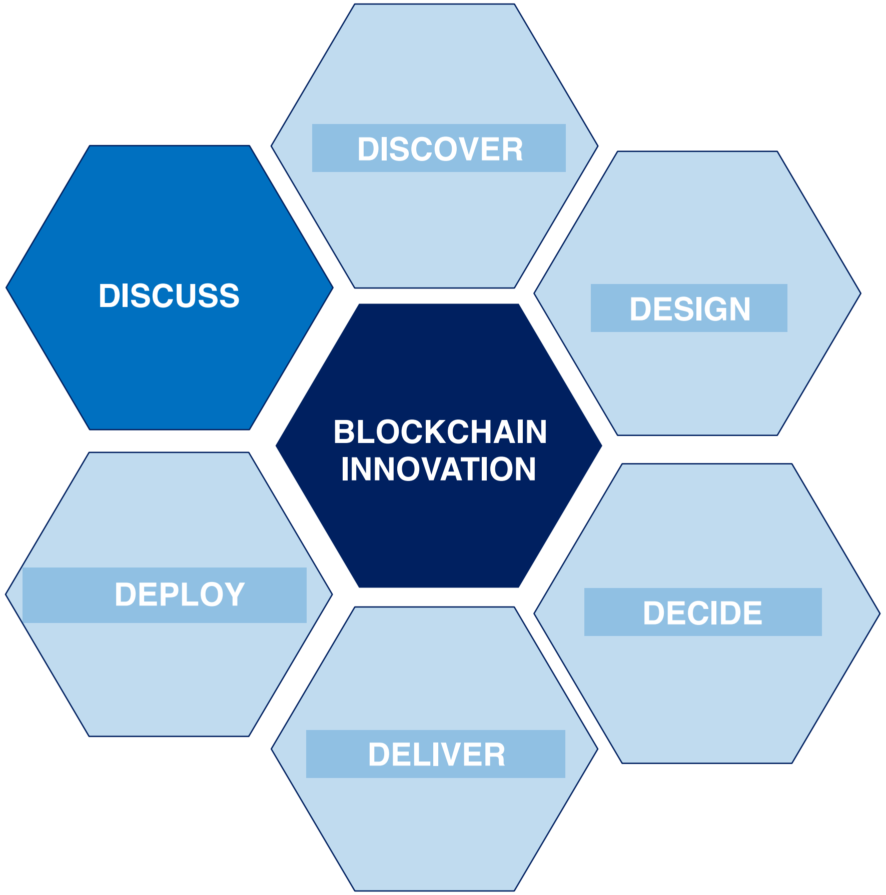
* Collect availability and response data from multiple points of presence in different geographic locations
* Detect any integration issues and conflicts early in the cycle.

Link to IBM Cloud Garage Method site for all practices

IBM Blockchain Innovation - Discuss

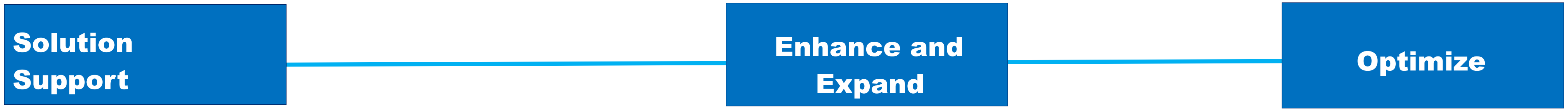
IBM Blockchain Innovation provides a practitioner friendly, tool based, light weight approach to deliver applications using Agile principles enabled by automation and DevOps.

- Collaborate with the client to optimize delivery processes, harvest assets/accelerators, add additional capabilities



WORK PRACTICE

Discuss – Enhance, Support & Optimize Solution



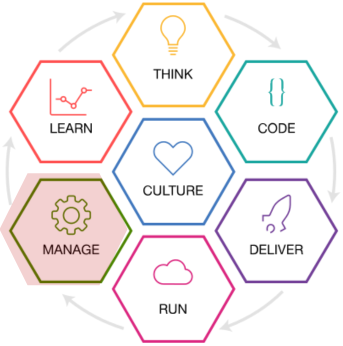
Solution Support

Enhance and Expand

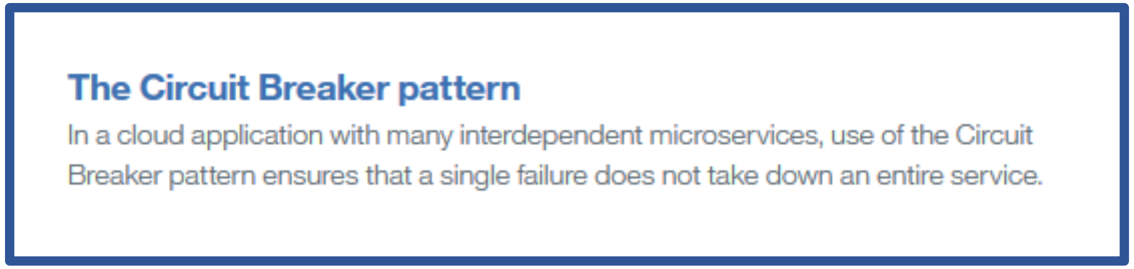
Optimize

	Solution Support	Enhance and Expand	Optimize
ACTIVITIES	▪ Transition from dev to support team ▪ Establish review and communication mechanism ▪ Fix defects ▪ Update architecture and Smart Contract documentations for application fixes	▪ Understand & Analyze solution enhancements ▪ Add advanced features / functionalities ▪ Implement Advanced features ▪ Expand Project scope by adding add-on Project services ▪ Add Project resources to 1) support additional development sprints, 2) support additional UAT sprints ▪ Develop roadmap for enhancements	▪ Assetization - harvest assets & accelerators ▪ Improve technical designs ▪ Upgrade tools & services ▪ Improve delivery procedures (service level, incident, configuration, change, upgrades) ▪ Optimize support model ▪ Knowledge management
DELIVERABLES	▪ Blockchain application with fixes, Updated application documentation	▪ Prioritized List of Enhancements, ▪ Implementation Roadmap, Proposal / PCR for enhancement ▪ Value Realization	▪ Optimized Roadmap, Lessons Learned, Re-usable Assets, ▪ Improved Delivery Procedures, Proposal / PCR for additional resources & add-on services
TOOLS, ASSETS		▪ Blockchain Solutioning tool, ▪ Project Planning Tool, ▪ Microsoft Word	▪ Blockchain Solutioning tool ▪ Microsoft Word,

Discuss – Key Practices to Manage Blockchain Solutions

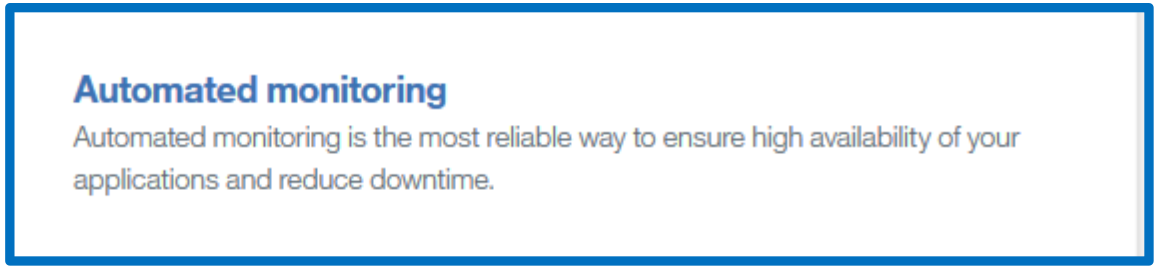


Manage



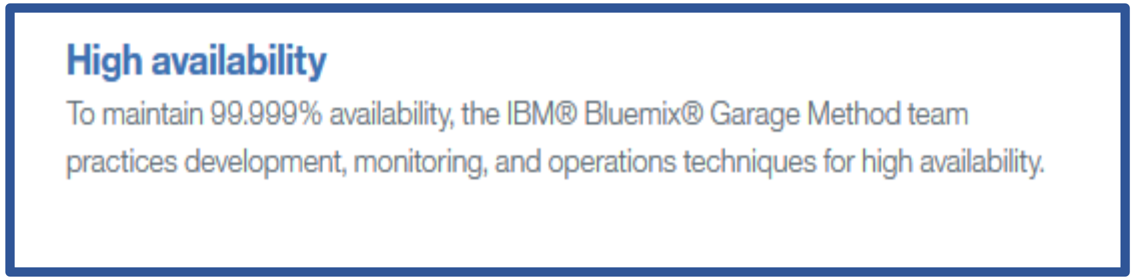
The Circuit Breaker pattern

In a cloud application with many interdependent microservices, use of the Circuit Breaker pattern ensures that a single failure does not take down an entire service.



Automated monitoring

Automated monitoring is the most reliable way to ensure high availability of your applications and reduce downtime.



High availability

To maintain 99.999% availability, the IBM® Bluemix® Garage Method team practices development, monitoring, and operations techniques for high availability.

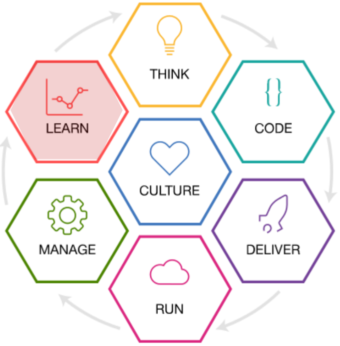
Incident Management

Record the problems, queries, and requests that application users submit
Solve the problems as quickly as possible to minimize impact on business.
Comply with agreed service levels and provide meaningful reports for management and other business processes.

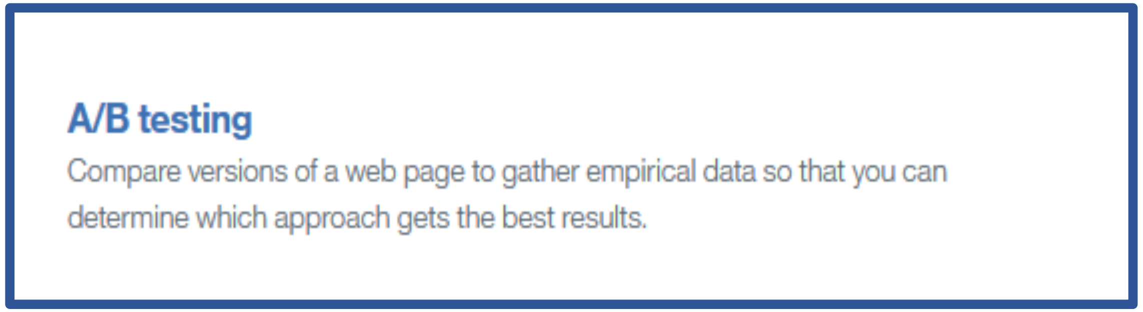
Link to IBM Cloud Garage Method site for all practices

WORK PRACTICE

Discuss – Key Practices to Learn, Revie & Revise

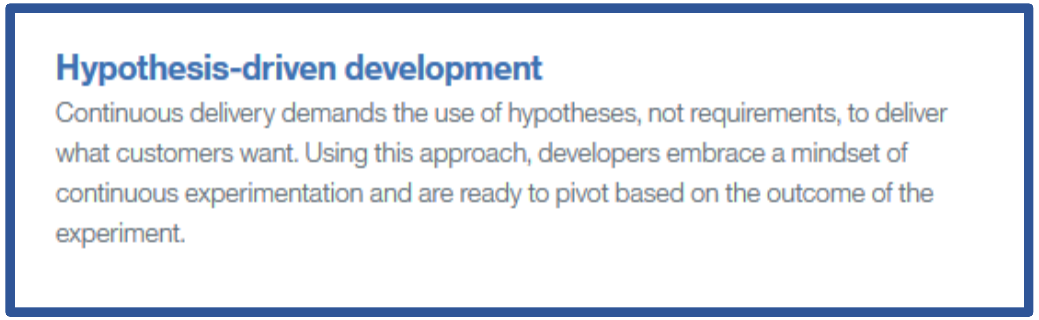


Learn



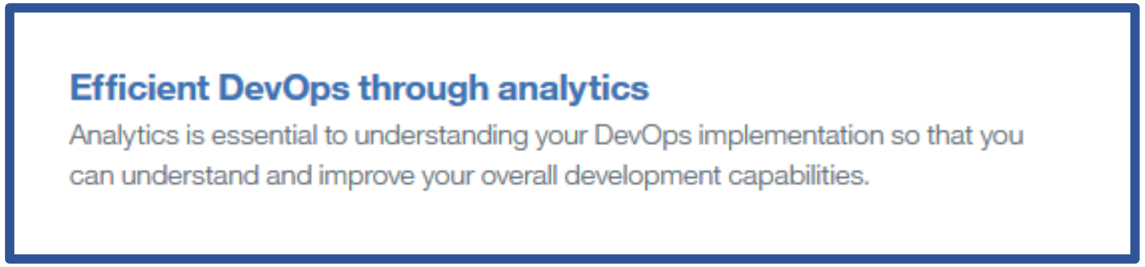
A/B testing

Compare versions of a web page to gather empirical data so that you can determine which approach gets the best results.



Hypothesis-driven development

Continuous delivery demands the use of hypotheses, not requirements, to deliver what customers want. Using this approach, developers embrace a mindset of continuous experimentation and are ready to pivot based on the outcome of the experiment.



Efficient DevOps through analytics

Analytics is essential to understanding your DevOps implementation so that you can understand and improve your overall development capabilities.

Learn from your users as they interact with the application - IBM Cloud Garage Method site