**What is ADC?**

ADC is an ecosystem which is transforming the way Application Development happens. It accelerates the development process through breathing a Lean Agile startup culture, Fluid workspace, Cloud ready Infrastructure, Rapid Application Development Accelerators, Integrated tool-chain, Cross functional Autonomous PODs. It fosters a culture of continuous innovation and delivers end-to-end full stack solutions.

1. ADC operates full-stack pods working in a Lean-Agile development outfit, independently managing their 2 week sprints to deliver System of Engagement projects.
2. A team of Business experts and UX Architects are seeded at client location to define Requirements, UX strategy & Planning and create technical solution architecture.
3. The development pods based offshore in India, work on the Design, Build, Unit Testing, Functional Testing and Virtualized SIT.
4. The certified build and automation scripts, are handed over to the client for SIT, UAT and to be deployed in production.

Within the ADC, all pods are supported by a gratis pool of architects and experts in disciplines like UX, Tooling, DevOps, Agile Coaches, etc. This gratis pool acts as synergists for seamless delivery at the ADC.

ADC is pre-equipped with all the required Delivery Tools, Solution Accelerators, Reference Architectures, and Reusable Components and has pre-configured Dev / Test Environments which makes it easier for the development pods to be productive from 1st hour of engagement initiation.

Coupled with TDD (Test Driven Development), BDD (Behavior Driven Development), Lean delivery and DevOps practices, The ADC also comprises of pre-configured Virtualized SIT Environment which impeccably simulates the actual SIT, resulting in the restriction of defect leakage. When it comes to engaging with variety of clients, ADC is flexible enough to work in a ‘Pay-Per-Screen’, ‘Fix Price’ or ‘Fixed Capacity’ model to suit different engagement model needs of our clients.

**ADC Accelerators Demo:**

**5 minutes: Floor Walk thru ADC walls**

**15-20 minutes: Demo of use case from Requirement to Deployment**

**5-10 minutes: Q&A**

**ADC Overview:**

* Presentation by opening up on ADC 6 key aspects:

**1) Real Estate,**

**2) Infrastructure,**

**3) Integrated Tool chain ,**

**4) DevOpe pipeline,**

**5) Rapid Application Development / Solution Accelerators,**

**6) Lean Agile,** Followed by How we operate, how we deliver and 40+ Technology stack coverage.

**Demo Order:**

**Digital BA – 5 minutes**

* **VVM – My3D** – Agile / Kanban story telling thru my3D application
* **BDD swift –** Capturing acceptance criteria right in the first place. It help to get the right requirements to build the right software. BDD editor helps to capture the requirement in Gherkin language (i.e.: Given When Then construct). It also creates the Feature and Step file using Cucumber / JBehave. It is integrated with Voice Bot assistance – Diana.
* **RDV** – Requirements, Wireframes, Visual design, Low / Hi fidelity Axure prototype along with real time collaborative feedback features.

**UI Developer – 5 minutes**

* **Appbuilder** – Drag and Drop interface and Responsive Web Application in browser

**App/ API Developer – 5 minutes**

* **AppSwift** for fullstack code generation
  + App Creation
  + Pipeline creation
  + Entity creation
* **SonarQube**, CDOE and DevOps pipeline
* **Deployed** app at IBM Bluemix cloud and open on mobile

**BDD swift**

Capturing acceptance criteria right in the first place. It help to get the right requirements to build the right software. BDD editor helps to capture the requirement in Gherkin language (i.e.: Given When Then construct). It also creates the Feature and Step file using Cucumber / JBehave. It is integrated with Voice Bot assistance – Diana.

Here is the information around ALEXA – Diana Bot:

ADC’s Diana - the voice enabled bot service that powers and provides capabilities that enable clients to interact with devices in a more intuitive way using voice. Clients can access these new skills on any Diana enabled device simply by asking Diana a question or making a command. Examples of these skills include the ability to understand ADC and it’s ecosystem, know more about any of the accelerators, answer general questions on ADC. Diana is built in the cloud, so it is always getting smarter. The more use cases we use Diana, the more she adapts to speech patterns, vocabulary, and personal preferences. It uses natural user interfaces, like those based on speech, represent the next major disruption in computing. One of such example is use of Diana in conjunction with Natural Language processing algorithms which gives a break thru in requirement capturing. As a pilot, we have enabled Diana with capabilities to capture the requirement from business and process in Gherkin language construct. i.e Given, when, then form. Thereafter it process this information to write BDD test cases and check into JIRA. Another use is to kick start the DevOps pipeline and keep use updated with status as it progresses. Thus Diana adds adding rich and intuitive experiences to your products and proves to be the right choice for you!

**Demos:**

1. **Digital BA - 5 mins**

Digital BA will demonstrate the PPT, where BA is going to talk about the USE CASE which we have choose for Guardian Insurance.

**Use Case:** Get a quote for Disability Income Insurance

Showing the Guardian Website and telling them it’s going to be responsive, which is going to work in multiple Browser and multi device environment.

And the use case will be implemented ADC way i.e.

1. How we will be doing it?
2. Secondly, how we are going to do it faster- starting from Requirement phase to cloud deployment.

So, that Client can relate the whole use case to their working site.

Implementation of the use case keeping in mind that team is distributed in different location like INDIA, USA, and MEXCICO.

To do it agile way we will be using Capgemini propriety In-house tool Virtual Visual Management (VVM). It’s **Digital Distributed Delivery (3D)** tool provides a platform to enable more effective interaction between distributed teams; The Virtual Visual Management (VVM) application uses synchronized web-based touch screen technology and can be accessed through any device to deliver information visually. It is used in review meetings and any interactive workshop.

VVM application is a visual way of capturing –

1. Team planning
2. Requirement, Gathering
3. User Acceptance Criteria
4. UI Creation
5. API level Creation
6. Deployment

And to cope with customer demand in an aligned and flexible way. The application provides a single interface to initiate, track and validate continuous improvement initiatives.

**Here is the information around ALEXA – Diana Bot:**

ADC’s Diana - the voice enabled bot service that powers and provides capabilities that enable clients to interact with devices in a more intuitive way using voice. Clients can access these new skills on any Diana enabled device simply by asking Diana a question or making a command. Examples of these skills include the ability to understand ADC and it’s ecosystem, know more about any of the accelerators, answer general questions on ADC. Diana is built in the cloud, so it is always getting smarter. The more use cases we use Diana, the more she adapts to speech patterns, vocabulary, and personal preferences. It uses natural user interfaces, like those based on speech, represent the next major disruption in computing. One of such example is use of Diana in conjunction with Natural Language processing algorithms which gives a break thru in requirement capturing. As a pilot, we have enabled Diana with capabilities to capture the requirement from business and process in Gherkin language construct. i.e Given, when, then form. Thereafter it process this information to write BDD test cases and check into JIRA. Another use is to kick start the DevOps pipeline and keep use updated with status as it progresses. Thus Diana adds adding rich and intuitive experiences to your products and proves to be the right choice for you!

**Rapid Design & Visualization (RDV) :** The ADC delivery pipeline starts with Rapid Design & Visualization (RDV), our proprietary methodology wherein our Business Domain and UX experts work with client stakeholders to capture, analyze, and document the requirements in form of high fidelity screens, and thus helps to freeze the requirements much faster (1.2x) than the usual process.

The User Stories, **BDD acceptance criteria**, and the screen simulations thus created in RDV are used in the next the stage where **‘BDD Swift’** helps to automatically generate the Executable Specifications of all test cases that propagate the TDD and BDD culture within the team. Subsequently, Pod members work on the Application Design and Domain Modeling which is then fed into ‘App Swift’ which automatically generates as much as, 30% of the application code that is based on **12-factor design principles** and is cloud-ready as well.

The app simulations created during RDV are fed into ‘App Builder’ which automatically creates the user interfaces based on HTML5 and adheres to the RWD principles. Upon completion of the application code by Pod, this application code is passed through ‘RoboQ’ for quality inspection to get rid of any ‘Technical Debt’ it may have incurred.

This process ensures that a very close eye is being kept on the overall quality of the software being developed.

This deployable code then processed by ‘Cloud Swift’ to deploy it into various clouds like PCF, AWS, etc. without making any changes to the original code. As part of the continuous delivery orchestration engine, ‘JIRA’ and ‘git’ provide a robust foundation for project tracking tool and code / document repository respectively.

**Axure RP** puts the power of prototyping into the hands of product managers, business analysts, and user experience professionals. Before writing a single line of code, solutions can be prototyped and validated by the people who best understand your business, products, and customers.

1. POWERFUL PROTOTYPES, WITHOUT CODING – Create simple click-through diagrams or highly functional, rich prototypes with conditional logic, dynamic content, animations, math functions, and data-driven interactions without writing a single line of code.
2. EASY TO SHARE – Click a button, and Axure RP will publish your diagrams and prototypes to Axure Share on the cloud or on-premises. Just send a link (and password) and others can view your project in a browser. On mobile devices, use a browser or the Axure Share App for iOS and Android.
3. DIAGRAMMING + DOCUMENTATION – Create flowcharts, wireframes, mockups, user journeys, personas, idea boards and more. Quickly drag and drop elements from built-in or custom libraries to create your diagrams. Then, style it with fills, gradients, line styles, and text formatting.
4. EASIER TEAM COLLABORATION – Axure RP allows multiple people to work on the same file at the same time, making it easier for your team to work together. Create a "Team Project" on Axure Share or SVN and use a check-in and check-out system to manage changes. Team projects also keep a history of changes with notes for each check-in.
   1. **UI Developer – 5 minutes**

**MVP – Minimum Viable Product:** A minimum viable product (MVP) is a development technique in which a new product or website is developed with sufficient features to satisfy early adopters. The final, complete set of features is only designed and developed after considering feedback from the product's initial users.

**Bootstrap** is a free and open-source front-end web framework for designing websites and web applications. It contains HTML- and CSS-based design templates for typography, forms, buttons, navigation and other interface components, as well as optional JavaScript extensions. Unlike many web frameworks, it concerns itself with front-end development only.

**Scaffolding:** Scaffolding is perfect for getting an application up quickly. In cases of standard applications, the scaffolding is perfect for getting an MVP up and running in little time at all, allowing you to focus on the core features of your application.

**AppBulider** is a Capgemini propriety tool which help us to create UI in a drag & drop manner. It's a responsive tool for easy development.

**Why AppBulider?**

There are plenty of frameworks and plugins available for Development but they are lack of automation accelerators. AppBulider offer is

1. It’s an automated accelerator works on scaffolding principle i.e. faster Design & Development.
2. 20 – 30% of auto code generation.
3. Responsive UI Interface for rapid development.
4. Reduced technical debt & improved code quality with all standards.

**AppBulider has three main section –**

1.       Left section - which offers 100 + UI components for UI design

2.       Center section – It’s a Canvas area for UI development, where you can easily drag & drop the components.

3.       Right section - Quick glance for developers, how UI design will look like in different gadget’s i.e. look and

                Feel.

**This tool offers –**

1. Easy copy/ paste of html code base or you can download complete code as a zip file. It's create an App, Which contain all your inner HTML, Views and test controls.
2. Responsive UI interface where you can easily change colors, look & feel as per client requirement.
3. Most important feature of this tools is Code which got generated is –

a.       HTML 5 Compliance

b.      Accessibility Compliance

c.       Semantic HTML5 Compliance

d.      And it’s based on MVC architecture.

**Generated Code base is –**

1.       HTML5

2.       CSS 3

3.       Angular 1.5

4.       Bower

5.       Bootstrap

6.       Grant

**For Test Controller -**

1.       Protector

2.       Karma

3.       Jasmine

4.       Selenium

**Technology which goes behind the AppBulider -**

1.       Yeomen - to write generator

2.       Grant - in terms of task automation

3.       Bower - in terms of dependency mgmt.

4.       Angular 1.5

5.       ECMAScript 6

**1.3 API Developer – 5 minutes**

**MVP – Minimum Viable Product:** A minimum viable product (MVP) is a development technique in which a new product or website is developed with sufficient features to satisfy early adopters. The final, complete set of features is only designed and developed after considering feedback from the product's initial users.

**Scaffolding:** Scaffolding is perfect for getting an application up quickly. In cases of standard applications, the scaffolding is perfect for getting an MVP up and running in little time at all, allowing you to focus on the core features of your application.

**AppSwift** is our one of the Solution Accelerator for rapid development with integrated DevOps tools that helps us generate a complete and modern Web app, unifying development until deployment to production servers which is automatically done by answering few questionnaires.

1. AppSwift generates high-performance and robust Java or .net stack on the server side with lot of open source technology to choose from.
2. A powerful workflow to build your application with template based code generation and unified packaging and bundling as Docker images which finally gets deployed to most widely adopted cloud platforms such as IBM Blue Mix, Amazon AWS, Pivotal cloud foundry and Microsoft Azure.
3. It has built in Unit Test generation templates along with integrated static code analysis factory which helps finding early code related concerns in development phase.

Let’s start creating the domain model for our sample guardian demo app which will be as per the wireframe from the BDD Swift capturing basic customer information. **“Generate Entities”** is a visual Entity builder which currently works with MySQL, oracle and PostgresSQL. It helps us creating the required entities which can be used to create the schema in the database.

“Create APP” is a visual App Builder which takes the inputs in the form of a visual questionnaire. By answering few questions we can generate the application using standardized templates, which generates the basic scaffolding with all necessary integration required for end to end development. These are highly configurable template driven code accelerators which helps in accelerating the environment setup and get the application development in a rapid phase.

Pipeline Builder helps us design various pipeline stages from Code compile to creating a dockerized container and finally pushing it to cloud container.

Once the Pipe line is built, we have the CDOE which helps us collaborating all the stages in one place where we can monitor the app build orchestration console logs.

Once the pipeline is in place, we can start the build phase and wait for the app deployment. We have prebuilt the demo application which is hosted in bluemix cloud which is ready and can be accessed at <https://guardian12.mybluemix.net/#/>

**What goes behind? (for Information purposes)**

**JHipster** is a free and open-source application generator used to develop quickly a modern web application using AngularJS and the Spring Framework. JHipster provides tools to generate a project with a Java stack on the server side (using Spring Boot) and a responsive Web front-end on the client side (with AngularJS and Bootstrap). The term 'JHipster' comes from 'Java Hipster', as its initial goal was to use all the modern and 'hype' tools available at the time. Today, it has reached a more enterprise goal, with a strong focus on developer productivity, tooling and quality.

**Major functionalities**

Generate a full stack application, with many options

Generate CRUD entities, directly or by scaffolding

Database migrations with Liquibase

NoSQL databases support (Cassandra, Mongo DB)

Elasticsearch support

Websockets support

Automatic deployment to CloudFoundry, Heroku, OpenShift

**Technology stack**

**On the client side:**

HTML5 Boilerplate

Twitter Bootstrap

AngularJS

Full internationalization support with Angular Translate

Optional Compass / Sass support for CSS design

Optional WebSocket support with Spring Websocket

**On the server side:**

Spring Boot

Spring Security

Spring MVC REST + Jackson

Monitoring with Metrics

Optional WebSocket support with Spring Websocket

Spring Data JPA + Bean Validation

Database updates with Liquibase

Elasticsearch support

Mongo DB support

Cassandra support

**Out-of-the-box auto-configured tooling:**

Yeoman

Grunt or Gulp.js

BrowserSync

Maven or Gradle