**Cloud computing**

-Delivery on demand computing resource that enables organisations to access and store information without managing their own physical devices or IT infrastructure.

Microsoft Azure

Amazon AWS: Focuses on startups

Google Cloud: Lot easier to learn and implement, but doesn’t have the massive data scalability of analytics.

Brief advantages: Lower latency, disaster recovery, rerouting, backups, automatic recovery, analytics of data. Faster to market, build incrementally. Allows replicated developing and testing environments to test new versions of applications. OR to test how SQL or DOS attacks would work without taking down actual app.

-**Faster time to market**: create template which any new stating proj can use  
-**Scalability and flexibility**: You can scale you resources/storage based on user base/requirements  
-**Cost savings**: Pay for what you use  
-**Better collaboration**: Distributed repos, continuous pushes/builds, work from anywhere. A lot of protocols and components are standards, so no need to build customer adapters, use what’s already there.  
-**Advanced security NB**: Time between finding vulnerability and a patch, can take very long (week+). With cloud, the updates to security rolls out to every environment. You aren’t in charge of having to do certain updates.  
-**Data loss prevention**:

**3 main uses**

Infrastructure as a service (Iaas): Environment in cloud you can deploy to

Platform as a service (Paas): Hardware and software resources needed. Hardware very expensive.

Software as a service (Saas): Using ready built service (eg point of sale) from cloud.

**Designs patterns and conventions in cloud**

**Availability**: You can reroute traffic if things go down, if high traffic. NB for banking apps as can never be down.

**Data Management**: For storing scalable data of all types. Documents, objects, ect. Migration, Easy to migrate to different platforms.

**Messaging**: Not just sending emails. Messaging queues from event based. So if one point of app goes down, you have a buffer queue that takes in those messages the app would have taken in, so when it goes up it takes in that buffer.  
-Now we can see queue is too big, you can scale up system temporarily to handle such.

**Resiliency**: Health checks, early warnings, monitoring, set up metrics and alerts.

**Design and implement**: leverage already built software

**CI/CD (Continuous Integration, Continuous Deplotment)**

Continuously:

Coding  
Building  
Testing  
Release  
Deploy  
Operate  
Monitoring  
Planning

Break down your work into small, digestible packages/features that allows easier testing, building and deploying weekly code.

Then you look back to what you’ve done, monitor it, plan more ect.

You evaluate what will add the most value this week. This continuos small pushes allows you to **adapt** and **not fall behind market.**

**Good source control is very important for this.**

-Whole team must be skilled in this.

**AI and Machine Learning**

-**Machine Learning:** Pattern Recognition used to evaluate massive amounts of data. Allows predictions to be made on this data.

**-Neural Networks:** Taking in large amounts of novel and unique sets of data. Classifying this data and evaluating it (making assumptions and giving insights). It takes learnings from many topics, so the more data comes in it is better able to predict from it.

**-Deep Learning:** Not just learning and alaysing models. Looking at the environment and inputs, and from those what tasks those inputs can do. Sentiment analysis.

Goals of AI:  
Planning:  
language processing: Take in complaints, frequently asked questions, chatbot. Taking this a bit further, running sentiment analyses to see who was happy with service, and those who weren’t can eg direct customer to person.

**NB:** AI is not just a box you buy and use. It is meant to augment all your other systems.

**Popular Technologies in South Africa**

**Python**: Many new people to language, as good for AI and scripting and data sciences in high demand.

**TypeScript**: See in Web

**C**#: Many windows applications before, but now very agnostics, run on cloud/any server/system. Backend services used in.

**Go:** Big data, understanding and collecting. Algorithms and maths.

**Javascript**: Java based web applications

**Java**: Still common for backend applications

**Rust**: Modelling language, building metrics, building insights, building data. Good for visual displays, is agnostics.

**Kotlin:** mobile side.

**Dart:** For small applications, widgets.

**C++:** Old systems, banks.

**Swift**: Istore, mobile apps. Faster rollouts as they can check the app faster (apple)

**YAML:** Environment configuration files. Read at runtime, not compiled in application.

**Software Industries**

Finance, SaaS, web dev, consulting, cloud based.

-IT now you have to talk to and learn other peoples expertise (eg medicine), to be able to make an app to better improve their field.

**Standbic Chaa App (What lecturer worked on)**

Digital banking app.   
Deployed in Kenya, where access to banking is not high.   
  
Had a 8 man team:

Business analysist  
Technology Analyst:  
-Do market research, get requirements and build estimates

2 android  
2 swift  
2 Dotnet backend

**6 main IT roles**

|  |  |
| --- | --- |
|  | Software Engineer |
| Attributes | Logical Thinker  Detailed Oriented Fast Learner |
| Responsibilities | Writing Code  Investigating bugs  Translating Business requirements into functional applications |
| Description | Code Specialist, solve puzzles, quality code, documentation, always learning. |

|  |  |
| --- | --- |
|  | Data Scientist |
| Attributes | Math giftset  Understands complex patterns  Passion for numbers |
| Responsibilities |  |
| Description |  |

-More internal role, not customer talking focused

|  |  |
| --- | --- |
|  | UX/UI |
| Attributes | Creative Strong Intuition Artistically gifted |
| Responsibilities | Wireframes, colours, looks and feels, validation messages, how customer sees messages |
| Description |  |

|  |  |
| --- | --- |
|  | Business/Systems analyst |
| Attributes | Good communication skills  Business minded  Always thinking ahead |
| Responsibilities |  |
| Description | Not very coding focused. Many meetings. Making sure app solves requirements. |

|  |  |
| --- | --- |
|  | QA engineer |
| Attributes | Inquisitive Detail attentive Organised |
| Responsibilities | Builds scripts to test apps |
| Description | Building automated test plans, load/stress testing environments. Making sure app wont be broken before going to production. |

|  |  |
| --- | --- |
|  | Infrastructure and Support |
| Attributes | Technically minded  Able to work under pressure  Passion for technology |
| Responsibilities | Setup and configure networks, user accounts. Diagnosing network issues, hardware and software issues |
| Description |  |

**Day of work**

Stand Up: 15 min, everyone up to date/whats happening

Sprint planning: What must be done in 2 weeks, the plan for it

Estimations: How complex is a part of work (ticket). As a senior vs junior will take different times

Code Reviews: Part of constant development process.

**Progress through career**

**Graduate/Junior:** Main responsibility is to learn and be committed. Need a senior/mentor. (0-3yr)

**Intermediate:** Can do work on their own. (3-6yr experience)

**Senior:** Expected to teach/guide others.

Below 3 in parallel/on same level

**Specialist Developer**

**Team Lead:** Anyone can be a team lead, depends on topic/focus of project

**Technical Lead:** Decisions on services used, what infrastructure, languages used, team makeup

Higher level

**Practice Head:** Expert in everything, hires and fires, commissions projects, has business plan