**Lean Software Development**

Lean Fundamentals

Kanban, Value Steam Mapping and Kaizen

Lean Startup

Design thinking

he course is designed to provide practical knowledge around how to apply lean methods and principles in software development.

This course assumes that you have at least 1-2 years of experience in the software development industry.

We **start the course** by learning **Lean Principles from Manufacturing.**

After that we will:

* Discuss how to apply Lean principles to software development.
* Dig deeper into Lean Software Development Principles based on the Lean Software Development books by Mary and Tom Poppendieck,

This module will set us up with a foundation for next three modules.

**NOTE: In this course I have used some external videos as these external videos provide excellent explanation on the subject. I intend to replace those videos once I have content to improve upon them.**

In the **second module**, we will learn about three very popular Lean techniques used in the software industry:

* Kanban
* Value Stream Mapping
* Kaizen

In the second module, you will also do a **peer-reviewed assignment where you will create a Value Stream Map from a fictional case study**.

In the **third module**, we will learn about **Lean Startup** based on the book called "The Lean Startup" by Eric Ries. It may appear that Lean Startup should only be applicable to startups but it is not true. The concepts of Lean Startup are being used quite widely in the software industry. I have included a video by Eric Ries that I would highly recommend you watch as he shares many real stories, provides valuable insights, and explains the thought process behind his theory.

In the **fourth module,**we will start with learning about the theory of **Design Thinking.**Then, we will learn a technique called "Lean Startup with Design Thinking" based on book called "User Story Mapping" by Jeff Patton. To learn his technique, we will take a fictitious example and go through each of the steps exactly as he describes.

Finally, in this module, you will do a  **peer-reviewed assignment where you will apply the technique learned in this module to a fictional case study**.

**What do we mean by Lean?**

It is a mindset – way of thinking, not a new process or model

Set of principles and ideas, lots of tools & techniques

Origin from the manufacturing world

Core focus on the customer, maximize value, minimal resource

Optimize the flow of products and services

Eliminate waste across the entire value stream

* Less time
* Less people
* Less cost
* Less defects

= happy customers and more profit

It is a way of thinking

Equally applicable to services and other areas

Lean = Lean manufacturing, lean software development, lean product development, lean management, lean startup

**5 principles of Lean:**

* Identify Value – from the customers perspective
* Map the Value Stream – map the steps in the value stream
* Create Flow – create a smooth flow
* Establish Pull – the right amount at the right time
* Seek Perfection – elimination of all waste in the value stream

Value  
Value Stream  
Flow  
Pull  
Perfection

Bringing Lean to Software Development

The 7 principles of software development

* Eliminate waste – defects or additional features
* Amplify learning
  + Create knowledge
* Defer commitment
  + Decide as late as possible
* Deliver fast
  + Deliver as FAST as possible
* Empower the team
  + Respect people
* Build quality in
  + Build integrity in
* Optimize the whole
  + See the whole

**Kanban**

Encourages team to optimize the current flow they have but putting WIP in place for each steps

**Principles**

* Start with what you do know
* Agree to pursue incremental evolutionary change
* Respect the current process, roles, responsibilities and titles

**Properties**

* Visualize the workflow
* Limit WIP
* Manage flow
* Make process policies explicit
* Improve collaboratively

**Leam Startup**

Not can it be built – SHOULD it be built

4 principles

Validated learning

Entrepreneurs are everywhere

Entrepreneurship is Management

Innovation accounting

Value stream mapping, Kaizen, Theory of constraints, Cost of Delay

Lean Principles for software Development

Eliminate Waste –

8 types of waste-

* Defects
* Over production
* Waiting
* Non Standard Processing
* Transportation
* Intellect
* Motion
* Excess inventory

Amplify Learning / Create Knowledge

What is it?

To increase the ability of a team to learn quickly and effectively – for the customer

User needs Solution Process

**Why does it matter?**

Software development is a creative process

* Solution is unknown
* User needs is unknown
* Problem itself is unknown

Try -> Learn

Ability to learn quickly and effective is a competitive advantage

**How do you amplify learning?**

**Iterations** – Short development cycles

**Synchronization** – Spanning application = limited scope implemented end to end by advance team

* Develop interfaces together -> develop components individually
* Daily builds and smoke tests – continuous code integration and automated tests

**Set based development** – development multiple options -> solution emerge

**Defer Commitments**

What is it?

Ability to make commitment as late as possible. Make decision when you have the information you need. Does not mean delay the development

Why does it matter?

* We have to make all sorts of decisions
* As project progresses we have more information
* The sooner we make a decision the more chances of it being wrong
* Costly

Delay decision until we have the information we need or it is really required

How do you defer commitment?

Concurrent vs Sequential development

Ability to postpone decision until last responsible moment or make it possible to make changes easily so changed decisions can be digested effectively

Loosely couple architecture

* Use modular design, encapsulation, separation of concern
* Use interfaces and abstraction
* Use parameters

Avoid repetition – change once and be done

Deliver fast compliments this principle

**Build Quality In**

What is it?

Don’t defer quality checks to a development phase. Instead build quality from the start and at every step. Instead of tracking defects try to prevent them

Defects = Waste

Easier to find and easier to fix

Measure true progress

Defect masking

How do you build quality in?

* Standards – reduce conversion waste – Have to be used to be useful
* Code reviews
* Pairing
* Mistake proofing -> Automation
* Automated tests
* Test driven development
* Continuous integration
* Awareness of importance of quality in team

**7 Key Principles of Lean Software Development:**

1. [Eliminate Waste](http://www.allaboutagile.com/2010/08/lean-principles-1-eliminate-waste.html)  
2. [Build Quality In](http://www.allaboutagile.com/2010/09/lean-principles-2-build-quality-in.html)  
3. [Create Knowledge](http://www.allaboutagile.com/2010/10/lean-principles-3-create-knowledge.html)  
4. [Defer Commitment](http://www.allaboutagile.com/2010/11/lean-principles-4-defer-commitment.html)  
5. [Deliver Fast](http://www.allaboutagile.com/2011/01/lean-principles-5-deliver-fast.html)  
6. [Respect People](http://www.allaboutagile.com/lean-principle-6-respect-people/)  
7. [Optimise The Whole](http://www.allaboutagile.com/lean-principle-7-optimise-the-whole/)

**Deliver Fast**

What is it?

Reduce development cycle time without compromising the quality

Why does it matter?

Customer like faster delivery

Compliments ‘Defer Commitment’ principle

Reduces risk / reduces waste

How do you deliver fast without compromising?

Reduce cycle time

* Cycle time = things in progress / Avg completion rate

Minimize number of items in process

Minimize size of things in progress

Limit work to capacity

Use pull scheduling

**Optimize the whole**

What is it?

Consider the complete system when optimizing rather than individual component – optimize the whole system rather then individual component

Why does it matter?

Avoid optimizing the wrong component

In most cases optimizing one component leads to surprising unintended consequences in other parts of system

A system is not just the sum of its parts it is the product of their interactions

How do you optimize the whole?

Limits to growth

Kanban – helps you identify issues in your software development process flow by limiting work in progress

**Kanban**

Kanban for software development

Set or properties and principals so long as there is a continuous flow.

Backlog – Software development pipeline – end result

Backlog is a list of work to be done

Backlog, Analyze, Develop, Rest, Release – all steps are put into columns, with Doing, Done breakdown within each

From the backlog each task with the breakdown is moved from Doing to Done as it progresses.

1 – Visualize your workflow

2 – Limit WIP (work in progress) to a set number so to prevent log jam in testing for example, once work is done if there are too many items in a WIP then it should be looked as to why, code is wrong, testing taking too long etc.

3 – Manage the flow, are there issues, is the code wrong, does it require more people.

4 – Make process policies explicit – When is something done to move to testing

Kanban Principles

* Start with what you do know
* Agree to pursue incremental evolutionary change
* Respect the current process, roles, responsibilities and titles

Kanban Properties

* Visualize the workflow
* Limit WIP – work in progress
* Manage flow
* Make process policies explicit
* Improve collaboratively

<https://www.planview.com/resources/articles/lean-metrics-improve-flow/>

Know the Economics of Your System

1. Work in Process or Work in Progress (WIP)

2. Queues

3. Blockers

4. Lead time and cycle time

5. Throughput

6. Little’s Law in practice

7. Cumulative flow diagrams

**Value Stream Mapping**

Map out activities of the process you want to analyse

Go to the place where work is happening

Identify value added work, non value added work

Who are involved, what is involved and how long it takes

Syntax & Terminology

Value stream

A = 5 mins

B = 5 days (non value added)

C

C = 1 hour

D – 1 day (non value added)

In this case we will show time in minutes

5 days (1 day is 8 hours) = 8\*60mins = 2400 mins

A = 5 minutes

B = 2400 mins

C = 60 mins

D = 480 mins

Process Cycle Efficiency = value added time / cycle time

Value added time = time spent in doing things that add value for the custome

Non value added time = Time spent in doing things that didn’t add value for the customer

Cycle time = total time take for value stream

Process Cycle Efficiency =

(5+60) / (5+60+2400+480) = 0.022 = 2.2%

Case Study

**Kaizen**

Translated the word kaizen means to “change for the better”. In The Kaizen Way you’ll learn exactly what kaizen is and how it’s helping organizations in industries such as manufacturing, office environments, and healthcare reduce costs and increase profits without resorting to massive layoffs. In fact, during The Kaizen Way you’ll hear how one company managed to increase net profits by close to 25% even though their sales were down by 25% the same year!

Breaking down the current process put it back together with impovements

Standard work Heijunka

The 3 Gen’s

Genb-a / Gem-ba – the actual place, where the actual work is done

Gen-butsu – actual parts

Gen-jitsu – Get the facts

5 Whys – helps identify the root cause

Peer-graded Assignment: Value Stream Map

**Assignment Topic:**

In this assignment, you will be asked to create a value stream map for the scenario below. You will then use it to compute the process cycle efficiency and make recommendations for improvements.

Wise Software Corp (WSC), a short-term contracted software development company, has the following process for serving its clients.

1. Clients first create account by providing basic profile information (10 min)
2. Clients wait for the approval of their create account request (avg 12 hrs)
3. Company's system runs in the night and sends email confirmation email (1 min)
4. Email sits in Clients email box (avg 12 hrs)
5. Client confirms their account by clicking on a link in email. (1 min)
6. Client submits a new project request on WSC's website portal (1 hr)
7. Request is waiting to be assigned to an account manager (12 hrs)
8. Company's Sr Account Manager looks at the pending requests every day and assign it to one of the account manager (1 min)
9. Request is waiting for account manager to looks at the request (12 hrs)
10. Account manager contacts the client to set up a meeting to get requirements (30 min)
11. Client and Account Manager waiting for the meeting (72 hrs)
12. Account Manager and Client meet to outline requirements(4 hrs)
13. Account Manager puts the request in queue to document (24 hrs)
14. Account Manager write up requirement and send this information to Company's Solution Architect (30 min)
15. The project is waiting for solution architect to provide input (48 hrs)
16. Solution Architect looks at and provide feedback to Account Manager (1 hr)
17. Solution Architect feedback waiting to be read by Account Manager(24 hrs)
18. Account Manager sets up another meeting with Client to discuss architecture and timeline(30 min)
19. Client and Account Manager waiting for the meeting (72 hrs)
20. Account Manager and Client meet to approve architecture (1 hr)
21. Account Manager sends the project request to development queue (30 min)
22. Project request waits in development queue (120 hours)
23. Project team completes the development and assign it to testing team (72 hrs)
24. Project request waits in testing team to work on it (120 hrs)
25. Testing team tests and approve it for release (24 hrs)
26. Project waiting for Account manager to send the request to client (12 hrs)
27. Account Manager sends the complete project deliverables to Client. (10 min)
28. Client uses the deliverables. (0 min)

Please calculate the Process Cycle Efficiency for this Value Stream? Please show your calculations.

Make a recommendation for improving this process. Please specify what impact it will have in terms of VSM step and process cycle efficiency. Please calcuate new Process cycle efficiency if recommendations are implemented (HINT: For recommendations, think about merging steps, removing un-necessary steps or recommending alternatives to improve some of the process)?

**Lean Startup Principles**

Origin of Learn Startup

Lots of startups -> many of them fail -> waste

Set of principles, practices & tools to help manage innovation

‘ A human institution designed to create new products and services under conditions of extreme uncertainty’

Entrepreneurship is management

Management = bureaucracy -> avoid management and the ‘just do it’ attitude -> need management but different kind of management that supports extreme uncertainty

**Measure Progress**

Production of high quality products -> Validated learning – scientifically validating each element of startup vision by running series of experiments -> innovation accounting

**Approach**

User research, complex plans and flashy launch -> Build / Measure / Learn (this is the lean cycle) -> this is the pivot of persevere

Product – Frequently, based on build/measure/learn cycle

Strategy – Less Frequently – Pivot or persevere

Vision – Rarely changes –

**Applicable to only startups?**

Equally applicable to enterprise, small or large, profit or non-profit, commercial or government organizations -🡪 learn startup ecosystem – trying to create new products and services under conditions of extreme uncertainty.

**Entrepreneurs are everywhere**

Intrapreneurs 🡪 entrepreneurs that are trying to build a startup inside and established organization

**Lean Startup – Validated learning and build-measure-learn) --- Eric Ries**

Learn Build

Assumption Experiment

Measure

Metric

Cycle should be completed as quickly as possible

Validated learning

Incremental and Iterative

Very short dev cycle

Very adaptive

**Pros**

Helps you learn faster and build the right product

Speed to market

**Cons**

May result in rework

Requires you to experiment iteratively with clients and users

**Uses**

Doubtful business case / user nee

Lots of high probability risks

**Innovation Accounting**

Do we really accounting – wouldn’t accounting be applicable once the startup is successful and established 🡪 Yes but a different kind of accounting, it helps you make the right decisions at the right time

3 Learning Milestones

Develop a baseline – establish real data to understand the current state of a company

Tuning the engine – experiments to try to move from current state to ideal state

Pivot or Persevere – Pivot if not making desired progress else, persevere

Develop a Baseline

Prototype

* Test multiple assumptions at a time
* Once assumption at a time – try the riskiest assumption first – these are known as a leap of faith assumption

Smoke test

* No real product – customers are given an ability to pre order the product that is not yet built
* This validates if customers are interested in trying product

Tuning the engine

Activities (product development, marketing or other initiatives) targeted towards improving the drivers of growth model

**The 3 A’s of metrics**

Actionable

* Must demonstrate the cause and effect
* Should not be vanity metric – vanity metics can lead to false conclusion and can mask the failure

Accessible

* Easy to understand
* Easy access to report/data – metrics part of the application

Auditable

* Data is credible
* Manually testable

Pirate Metrics

Acquisition – How do users find you

Activation – Do users have a great 1st experience

Retention – Do users come back

Revenue – How do you make money

Referral – Do users tell others

**Pivot or Persevere**

Startups runway

* Amount of time remaining to either lift off or fail
* Amount of money remaining /monthly burn rate
* Extend by getting more money or cut costs

Startups runway – true definition

* Number of pivots remaining
* Increase your runway by reducing time for each pivot

Pivots require courage

* Vanity metrics
* Incorrect hypothesis
* Afraid to not get the chance to prove

Pivot or Persevere meeting

* Schedule in advance

Types of Pivots

* Zoom in – focus on 1 feature
* Zoom out – expanded to include more features
* Platform
* Customer segment
* Customer need
* Channel of delivery
* Technology

<https://www.youtube.com/embed/FRR_jEDAfoo?rel=0&autoplay=1&t=4m32s>

**Design Thinking**

Origin of design thinking

Design is a way of thinking 🡪 method of creative action 🡪 IDEO – Commercial application of design thinking

* Design thinking is a process for creative problem solving
* Methodology for creative and practical wicked problem solving
* Design thinking is a human-centered approach to innovation that draws from the designers toolkit to integrate the needs of people, the possibilities of technology and the requirements for business success

Desirability (Human)

Viability (Business)

Feasibility (Technical)

The cross of all of these is innovation

Empathize – Develop a deep understanding of the challenge

Define – Clearly articulate the problem you want to solve

Ideate – Brainstorm potential solutions, select and develop your solution

Prototype – Design a prototype or series of prototypes to test all or part of your solution

Test – Engage a continuous short cycle innovation process to continually improve your design

**Empathize**

Why?

* Prevent bias / filter
* They are not our problems
* Identify real problem

How?

* Go where the users are
* Talk directly to customer and users
* Watch them work, work with them
* Identify pains / needs

**Define**

Why?

* Keep focus
* Define the target group
* Define the real problem

How?

* Put everything you observed on board
* Share stories
* Distill learning
* Define the user / target grou
* Define / redefine the problem statement
* Focus on one of few problems

**Ideation**

Why?

* Don’t lock in too early
* Defer commitment, amplify learning
* Find innovative solution

How?

* Come up with multiple solutions
* Go beyond rational thinking
* Two rules – visualize, don’t sit
* Select few

**Prototype & Test**

Why?

* Validating ideas (quick and inexpensive)

How?

* Physical interactive prototype
* Right fidelity
* Prototype with defect is ok – main idea is to validate if it solves the problem
* Done defend the idea – note doen and move on

**Lean Startup with Design Thinking**

Build, Measure, Learn

Empathize, Define, Ideate, Prototype, Test

Big idea 🡪 DEFINE 🡪 Assumptions 🡪 Riskiest Assumptions 🡪 Prototype (learn as quickly as possible) 🡪 Focused Test (customer intercept, observe, interview) 🡪 Learning (celebrate learning, rethink solutions and assumptions) 🡪 the cycle continues Assumptions

**Identifying and Classifying Assumptions**

Listing assumptions

* Collective brainstorming (tech, business, stakeholders – everyone)
* What are we assuming about our users / customers
* What are we assuming about our solutions
* What must happen for our solution to be successful after it is released?
  + Will users engage
  + Will the recognize value
  + Will they be able to use the solution
  + Will have they have what they need
* Start with High probability that it will be wrong, high impact on solution
* 2 dimensions
  + Probability that it will be wrong
  + Impact on our solution if it is wrong
* 4 variations
  + Low probability that it will be wrong, low impact on solution
  + Low probability that it will be wrong, high impact on solution
  + High probability that it will be wrong, low impact on solution
  + High probability that it will be wrong, high impact on solution
* Start with High probability that it will be wrong, High impact on solution

College wants to improve image

Appoint CTO to run it

CTO collects team – proposal – all books digitized, every student gets a tablet, costs included in college fees

Befits – much better reading experience, no more carrying books, better for takeing notes, increase school reputation as tech savvy

What assumptions

Is it a problem for students to carry around books? – low probability of being wrong, low impact

Tablets will provide better reading experience “who doesn’t like gadgets’ – high probability of it being wrong, high impact

Students will be willing to pay for tablet cost – high probability of it being wrong, high impact

Prospective students will have favourable rating for college if we provide digitized books – high probability of it being wrong, low impact

**Prototype and Test**

Listing potential ideas

* Collective brainstorming (tech, business, stakeholders – everyone)
* Customer intercepts – go where users are
* Types of tests
  + Interview
  + Pitch your solutions and get a reaction
  + Prototype: different degree of fidelity – paper to real application / real data
  + Observe
  + Entice user on existing platform – feature-fake

Classifying MVP Tests / Experiments

* 2 dimensions
  + Cost / effort of conducting test
  + Quality of data collected from test
* 4 variations
  + Low cost, low quality
  + Low cost, high quality
  + High cost, low quality,
  + High cost, high quality
* Start with low cost, high quality of data

Is it a problem for students to carry around books? –

Tablets will provide better reading experience “who doesn’t like gadgets’ – observe & interview – high cost, high quality

campus survey – low cost, low quality

Students will be willing to pay for tablet cost – campus survey – low cost, low quality

A video & signup page – low cost, high quality

Prospective students will have favourable rating for college if we provide digitized books – A/B testing on college admission – high cost, high quality

Survey during open house for prospective clients – low cost, low quality

**Learning From Tests**

* Collective brainstorming (tech, business, stakeholders – everyone)
* Affinity mapping
* Consolidate learning
  + Invalidated assumptions
  + Validated assumptions
  + New ideas / questions / assumptions

**Assignment:**

In this assignment you will apply some of the Design Thinking techniques (identify assumptions and risks, select the riskiest assumptions, and write MVP tests for the riskiest assumptions) to a fictitious situation. Feel free to make reasonable assumptions about user needs and additional functionality required to satisfy user needs.

Please read the attached case study before you proceed further.

Case​ ​Study:​ ​Remote​ ​Deposit​ ​Capture

1. Recently,​ ​several​ ​banks​ ​have​ ​started​ ​offering​ ​customers​ ​remote​ ​deposit​ ​capture.​ ​With this​ ​new​ ​service,​ ​customers​ ​do​ ​not​ ​have​ ​to​ ​physically​ ​go​ ​to​ ​banks​ ​or​ ​ATM​ ​machines​ ​to deposit​ ​checks​ ​anymore.​ ​Instead,​ ​they​ ​can​ ​send​ ​checks​ ​as​ ​a​ ​scanned​ ​image​ ​through​ ​an Internet​ ​portal​ ​provided​ ​by​ ​the​ ​bank.
2. This​ ​technology​ ​can​ ​save​ ​banks​ ​and​ ​customers​ ​time​ ​and​ ​money​ ​making​ ​the transactions.​ ​Blue​ ​Bank​ ​is​ ​considering​ ​implementing​ ​this​ ​new​ ​service.
3. To​ ​use​ ​it,​ ​customers​ ​need​ ​a​ ​remote​ ​capture​ ​account​ ​with​ ​Blue​ ​Bank.​ ​In​ ​addition,​ ​they can​ ​either​ ​download​ ​an​ ​app​ ​on​ ​their​ ​phone​ ​or​ ​they​ ​need​ ​to​ ​install​ ​a​ ​browser​ ​plugin​ ​to use​ ​any​ ​general​ ​purpose​ ​scanner​ ​attached​ ​to​ ​their​ ​computer.​ ​The​ ​customer​ ​will​ ​be prompted​ ​to​ ​install​ ​the​ ​browser​ ​plugin​ ​when​ ​customer​ ​tries​ ​to​ ​deposit​ ​checks​ ​using bank’s​ ​internet​ ​portal.
4. Once​ ​the​ ​account​ ​is​ ​established​ ​in​ ​the​ ​system​ ​and​ ​customer​ ​has​ ​setup​ ​the​ ​mobile​ ​phone or​ ​browser​ ​plugin,​ ​customers​ ​will​ ​be​ ​able​ ​to​ ​scan​ ​all​ ​of​ ​their​ ​checks​ ​anytime​ ​and anywhere​ ​by​ ​using​ ​the​ ​bank​ ​mobile​ ​app​ ​or​ ​by​ ​accessing​ ​the​ ​bank’s​ ​Internet​ ​portal, logging​ ​in,​ ​and​ ​scanning​ ​the​ ​checks.
5. The​ ​service​ ​should​ ​be​ ​as​ ​easy​ ​as​ ​sending​ ​an​ ​attachment​ ​in​ ​an​ ​email.​ ​Of​ ​course,​ ​this new​ ​application​ ​has​ ​to​ ​be​ ​very​ ​reliable,​ ​secure,​ ​and​ ​easy​ ​to​ ​use.​ ​It​ ​must​ ​be​ ​integrated into​ ​the​ ​current​ ​Blue​ ​Bank​ ​Web​ ​site,​ ​and​ ​the​ ​Web​ ​site​ ​must​ ​also​ ​provide​ ​the​ ​ability​ ​for customers​ ​to​ ​purchase​ ​the​ ​general​ ​purpose​ ​pre-tested​ ​scanners.
6. Blue​ ​Bank​ ​will​ ​set​ ​up​ ​the​ ​scanner-purchasing​ ​ability​ ​with​ ​several​ ​appropriate​ ​hardware vendors​ ​and​ ​sell​ ​the​ ​devices​ ​at​ ​its​ ​physical​ ​banks​ ​as​ ​well.
7. Blue​ ​Bank​ ​is​ ​not​ ​sure​ ​yet​ ​what​ ​to​ ​charge​ ​for​ ​the​ ​scanners​ ​or​ ​service.​ ​The​ ​Website​ ​will also​ ​provide​ ​online​ ​technical​ ​support​ ​and​ ​instructions​ ​showing​ ​customers​ ​how​ ​to​ ​setup and​ ​use​ ​the​ ​new​ ​scanner​ ​and​ ​service.​ ​Support​ ​will​ ​be​ ​provided​ ​24/7​ ​via​ ​the​ ​Web​ ​site and​ ​telephone.
8. Several​ ​managers​ ​at​ ​Blue​ ​Bank​ ​are​ ​not​ ​sure​ ​that​ ​they​ ​should​ ​invest​ ​in​ ​remote​ ​deposit capture.​ ​The​ ​bank​ ​prides​ ​itself​ ​on​ ​being​ ​customer-oriented,​ ​having​ ​several​ ​large​ ​bank buildings,​ ​ATMs​ ​and​ ​remote​ ​bank​ ​sites​ ​in​ ​many​ ​grocery​ ​stores,​ ​primarily​ ​throughout​ ​the Midwest.​ ​Many​ ​of​ ​its​ ​customers​ ​have​ ​banked​ ​with​ ​Blue​ ​Bank​ ​their​ ​whole​ ​lives,​ ​but​ ​less than​ ​half​ ​of​ ​them​ ​use​ ​any​ ​of​ ​its​ ​current​ ​Internet​ ​banking​ ​services.
9. The​ ​new​ ​VP​ ​of​ ​IT​ ​was​ ​hired​ ​to​ ​prepare​ ​Blue​ ​Bank​ ​for​ ​the​ ​future​ ​and​ ​to​ ​reduce​ ​the​ ​loss​ ​of younger​ ​customers​ ​who​ ​have​ ​moved​ ​to​ ​banks​ ​that​ ​are​ ​more​ ​progressive.

10. **Initial**​ ​**estimates**​​ ​suggests​ ​that​ ​the​ ​project​ ​will​ ​take​ ​6​ ​months​ ​to​ ​complete​ ​and​ ​cost​ ​about $500,000.​ ​Annual​ ​operating​ ​costs​ ​would​ ​be​ ​about​ ​$150,000​ ​per​ ​year​ ​for​ ​three​ ​years. Estimated​ ​benefits​ ​are​ ​about​ ​$400,000​ ​per​ ​year​ ​for​ ​three​ ​years.

11. Harold​ ​Johnson,​ ​the​ ​project​ ​sponsor,​ ​had​ ​told​ ​you​ ​that​ ​​**you**​ ​**will**​ ​**be**​ ​**leading**​ ​**this**​ ​**project** and​ ​​​it​ ​is​ ​important​ ​to​ ​finish​ ​on​ ​time​ ​and​ ​within​ ​budget,​ ​but​ ​it​ ​is​ ​even​ ​more​ ​important​ ​to provide​ ​a​ ​high-quality​ ​service​ ​and​ ​avoid​ ​any​ ​potential​ ​technical​ ​or​ ​user​ ​problems​ ​after implementation.​ ​Blue​ ​Bank​ ​wants​ ​to​ ​attract​ ​new​ ​customers​ ​with​ ​remote​ ​deposit​ ​capture and​ ​avoid​ ​losing​ ​current​ ​customers,​ ​so​ ​quality​ ​and​ ​customer​ ​service​ ​is​ ​key.

12. **The**​ ​**team**​​ ​to​ ​work​ ​on​ ​this​ ​project​ ​has​ ​4​ ​developers,​ ​2​ ​testers.​ ​Most​ ​of​ ​the​ ​developers​ ​and testers​ ​has​ ​offices​ ​on​ ​different​ ​floors​ ​of​ ​the​ ​building​ ​and​ ​generally​ ​works​ ​in​ ​their​ ​office.

13. The​​ ​**team**​ ​**members**​​ ​have​ ​some​ ​​**experience**​ i​​ n​ ​web​ ​development​ ​but​ ​none​ ​in​ ​mobile development.​ ​Nobody​ ​on​ ​team​ ​has​ ​performance​ ​testing​ ​experience​ ​or​ ​User​ ​experience expertise.​ ​Also,​ ​since​ ​this​ ​is​ ​the​ ​first​ ​agile​ ​project​ ​for​ ​the​ ​team,​ ​they​ ​are​ ​not​ ​familiar​ ​with engineering​ ​practices​ ​like​ ​automated​ ​testing,​ ​continuous​ ​integration,​ ​etc.

14. **The**​ ​**Business**​ ​**stakeholders**​​ ​who​ ​wrote​ ​the​ ​high​ ​level​ ​vision​ ​for​ ​the​ ​mobile​ ​application are​ ​going​ ​to​ ​help​ ​refine​ ​or​ ​define​ ​the​ ​user​ ​needs.​ ​They​ ​currently​ ​sit​ ​in​ ​different​ ​building. They​ ​are​ ​working​ ​on​ ​writing​ ​detailed​ ​requirements​ ​and​ ​planning​ ​to​ ​hand-off​ ​to​ ​you​ ​in​ ​2 weeks.

1. Your​ ​peers​ ​have​ ​suggested​ ​that​ ​you​ ​want​ ​to​ ​be​ ​sure​ ​that​ ​the​ ​project​ ​will​ ​please​ ​key stakeholders,​ ​in​ ​particular​ ​Harold,​ ​the​ ​project​ ​sponsor,​ ​and​ ​Tricia,​ ​the​ ​VP​ ​of​ ​Marketing. Both​ ​of​ ​these​ ​senior​ ​managers​ ​are​ ​very​ ​aware​ ​of​ ​customers’​ ​needs,​ ​so​ ​it​ ​is​ ​important that​ ​the​ ​new​ ​service​ ​is​ ​stable,​ ​secure,​ ​and​ ​easy​ ​to​ ​use.
2. They​ ​both​ ​want​ ​the​ ​remote​ ​deposit​ ​capture​ ​capability​ ​to​ ​be​ ​ready​ ​as​ ​soon​ ​as​ ​possible​ ​so the​ ​company​ ​can​ ​start​ ​promoting​ ​this​ ​new​ ​service.

17. **High**​ ​**Level**​ ​**Vision**​:​ ​Here​ ​are​ ​the​ ​High​ ​Level​ ​Vision​ ​for​ ​the​ ​Remote​ ​Deposit​ ​Capture: **18.** Using​ ​the​​ ​**new**​ ​**mobile**​ ​**app,**​ ​​Blue​ ​Bank​ ​customers​ ​can:

1. Download​ ​an​ ​app​ ​for​ ​iPhone,​ ​android​ ​phones,​ ​windows​ ​phones
2. Login​ ​in​ ​the​ ​app​ ​using​ ​their​ ​internet​ ​portal​ ​account​ ​or​ ​identify​ ​themselves​ ​using

other​ ​authentication​ ​methods.

1. View​ ​balance​ ​of​ ​their​ ​accounts
2. Deposit​ ​checks​ ​to​ ​deposit​ ​account
   1. take​ ​secure/fraud​ ​proof​ ​pictures​ ​of​ ​checks​ ​(back​ ​and​ ​front)
   2. enter​ ​details​ ​of​ ​the​ ​check​ ​-​ ​like​ ​amt
   3. Select​ ​account​ ​to​ ​deposit
   4. validate​ ​check​ ​image​ ​against​ ​the​ ​entered​ ​details
   5. deposit​ ​the​ ​e-check​ ​into​ ​the​ ​bank​ ​account
3. Provide​ ​opportunity​ ​for​ ​bank​ ​to​ ​share​ ​new​ ​offers​ ​and​ ​promotions​ ​to​ ​customers.

**19. New**​ ​**browser**​ ​**plugin**​ ​​should​ ​support:

1. Secure/fraud​ ​proof​ ​scanning​ ​of​ ​checks
2. Support​ ​IE,​ ​Firefox,​ ​Chrome​ ​and​ ​Safari
3. Add​ ​support​ ​for​ ​pre-approved​ ​supported​ ​scanners.
4. Work​ ​with​ ​other​ ​scanners​ ​that​ ​supports​ ​ISO​ ​protocols​ ​for​ ​scanners.

e. Work​ ​seamlessly​ ​with​ ​pre-approved​ ​supported​ ​scanner  
20. **Changes**​ ​**to**​ ​**bank’s**​ ​**internet**​ ​**portal**​​ ​will​ ​allow​ ​Blue​ ​Bank​ ​customers​ ​to:

1. Create​ ​/​ ​update​ ​/​ ​delete​ ​deposit​ ​account
2. Scanning​ ​checks​ ​using​ ​the​ ​browser​ ​plugin​ ​(back​ ​and​ ​front)
3. Deposit​ ​e-check​ ​to​ ​deposit​ ​account

* ○  Enter​ ​details​ ​of​ ​the​ ​check​ ​-​ ​like​ ​amt
* ○  Validate​ ​check​ ​image​ ​against​ ​the​ ​entered​ ​details

Identify some of the assumptions made about the success of remote deposit capture functionality. Assess the risk of each assumption affecting our product by classifying the assumptions into following four categories:

1. High impact if wrong, High Probability of it being wrong
2. High impact if wrong, Low Probability of it being wrong
3. Low impact if wrong, High Probability of it being wrong
4. Low impact if wrong, Low Probability of it being wrong

For each assumption, provide the reason why you categorized the assumption into the particular category.

For any **three**assumptions you identified, please list the MVP Tests or Minimum Viable Experiments you can conduct to validate the assumptions? Please categorize your MVP Tests according to the following categories. It is okay to have 1 or 2 categories with no tests.

1. High Cost, High Quality Data (tests that will be costly to conduct, but will result in high quality data)
2. High Cost, Low Quality Data (tests that will be costly to conduct and will result in low quality data)
3. Low Cost, High Quality Data (tests that will be inexpensive to conduct, but will result in high quality data)
4. Low Cost, Low Quality Data (tests that will be inexpensive to conduct and will result in low quality data)