



Software Innovation and R&D Management

Project Introduction

2019-03-13



上海交通大学
SHANGHAI JIAO TONG UNIVERSITY

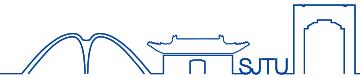


Project

- Requirements:
 - Carry out a idea which is validated with the methodology mentioned in this course
 - At least 2 rounds of customer validation
 - Ignore the technology issue
 - Implement your solution
 - At least 1 round of solution validation
 - Cover the cloud native domains
 - Use a git workflow
 - Operations
 - Use GitHub to record the whole process of your project
 - Prepare a PPT for presentation



Project



- Submission
 - At the end of semester
 - Deliverables
 - Code
 - Any other docs which can prove your performance should be included in this repo
 - Markdown file
 - PPT



Project



- Check points
 - 5th week
 - Team work environments
 - Github repo
 - Git workflow
 - CI/CD
 - Hypothesis
 - 9th week
 - Customer validation
 - Propose your solution
 - Solution validation



Project



- Check points
 - 13th week
 - Demo of your solution
 - 18th week
 - Operations



Software Innovation and R&D Management

Product Thinking II

2019-03-13



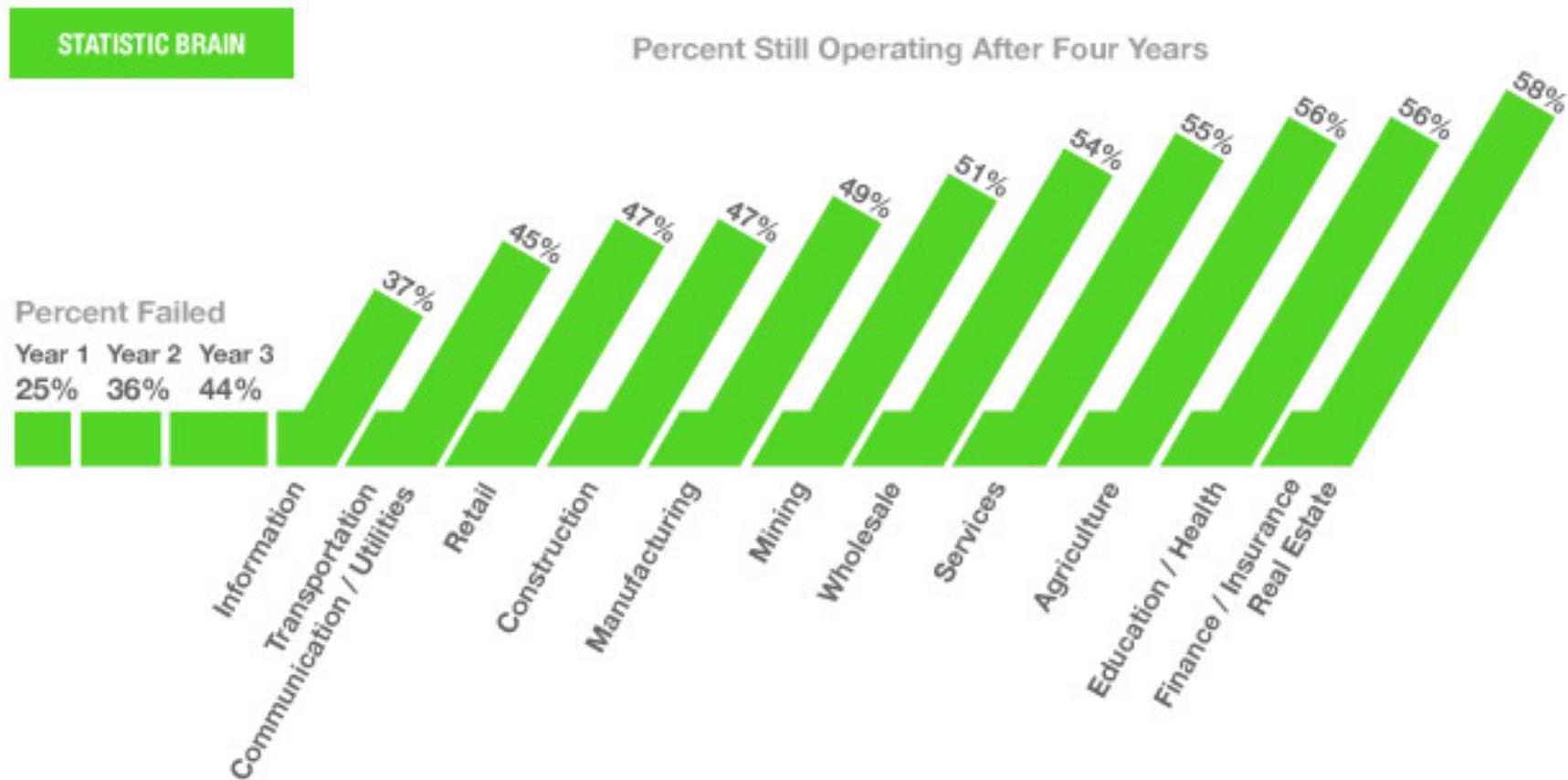
上海交通大学
SHANGHAI JIAO TONG UNIVERSITY



About startup

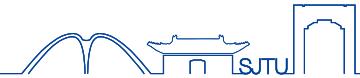


▪ In America





About startup



- In China
 - Average life time: 13.47 months
 - Ranking
 - Mobile internet (26%)
 - Electronic Commerce (25%)
 - Social (19%)

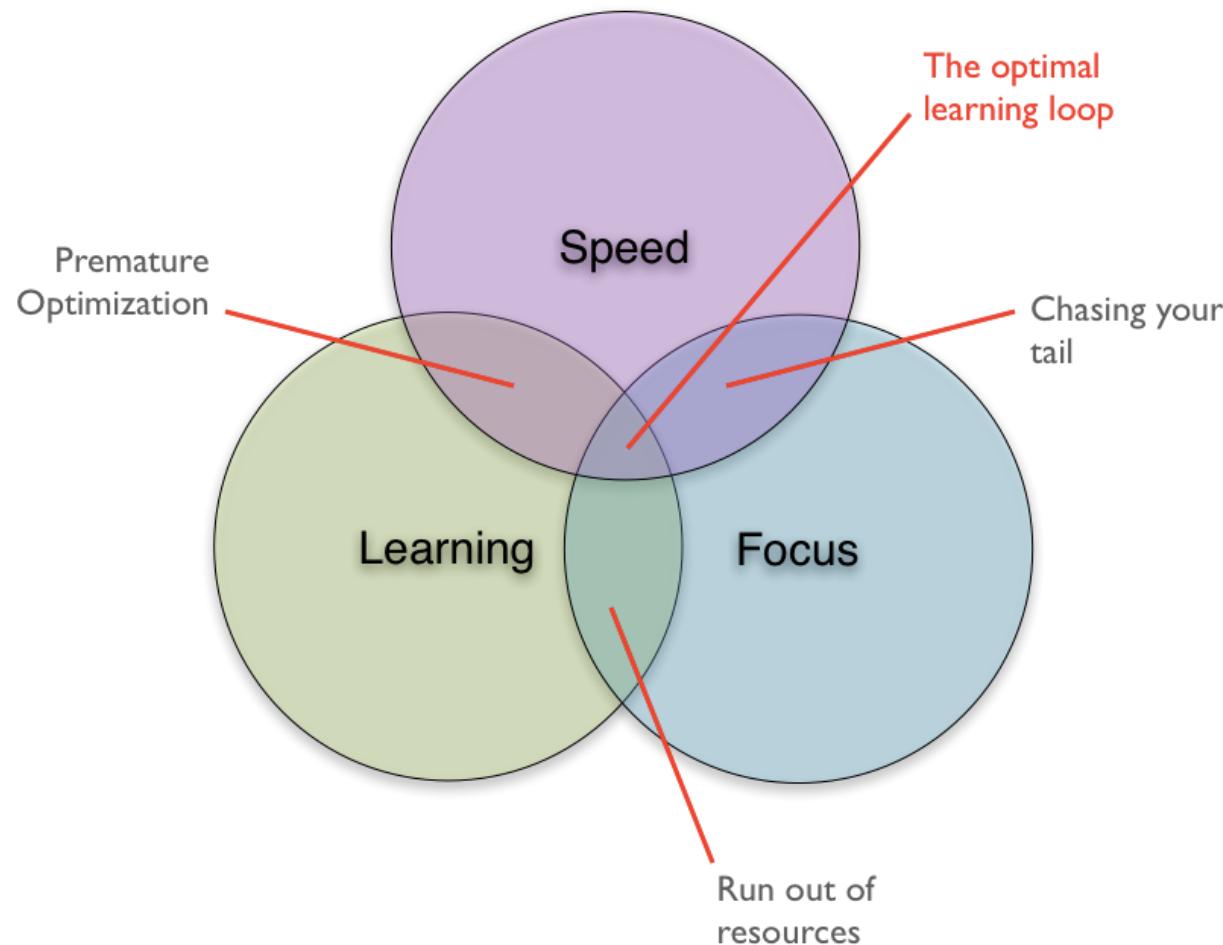


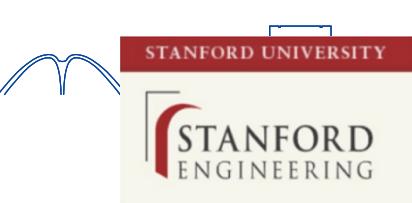
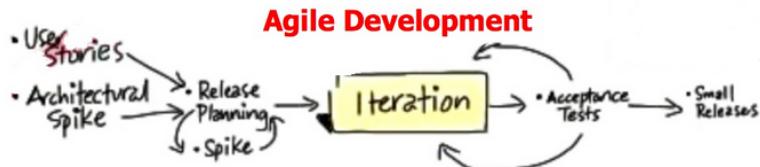
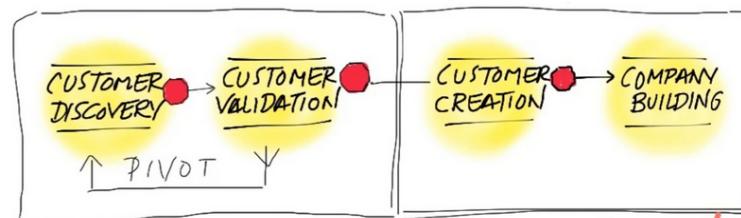
Why

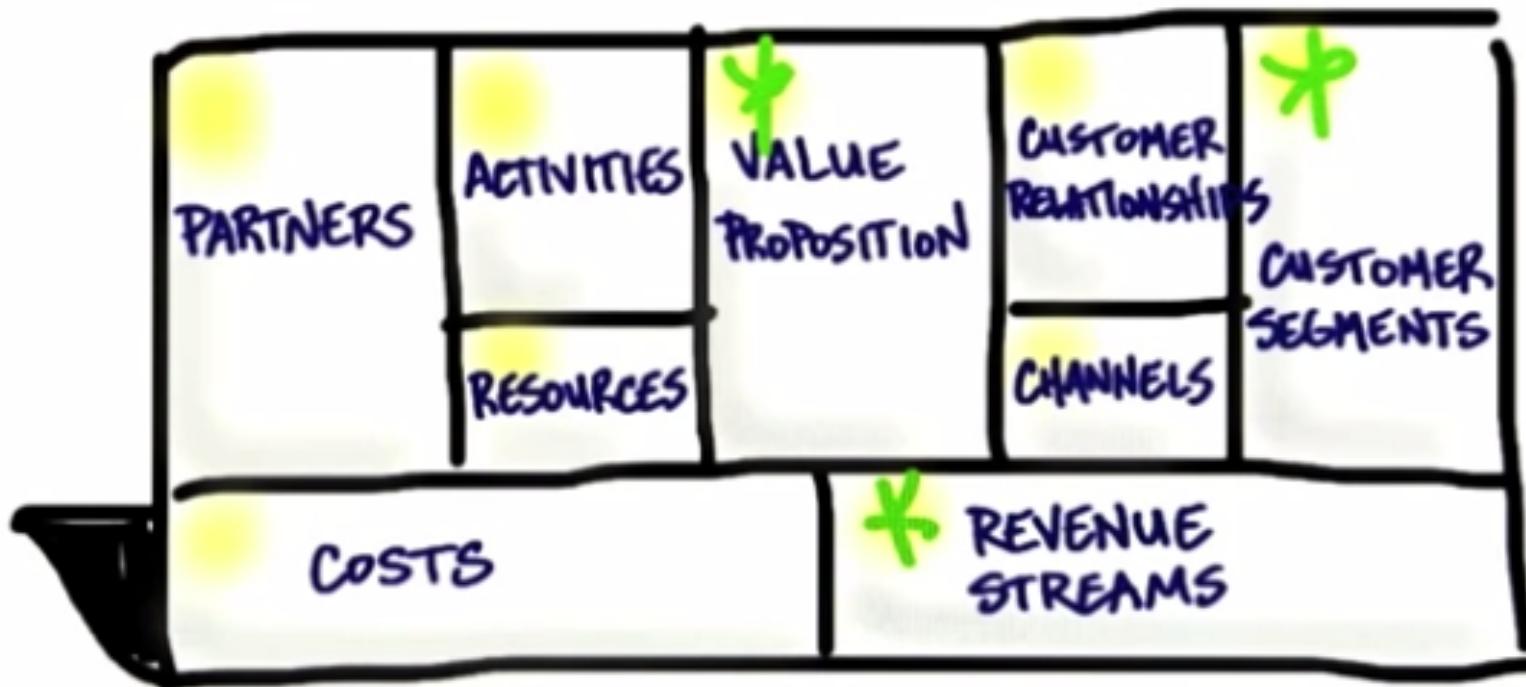
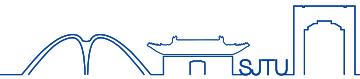


- Key factors
 - Focusing
 - Ambition
 - Weak Management Ability
 - Too slow
 - Focusing technology
- **Better product, few customers**



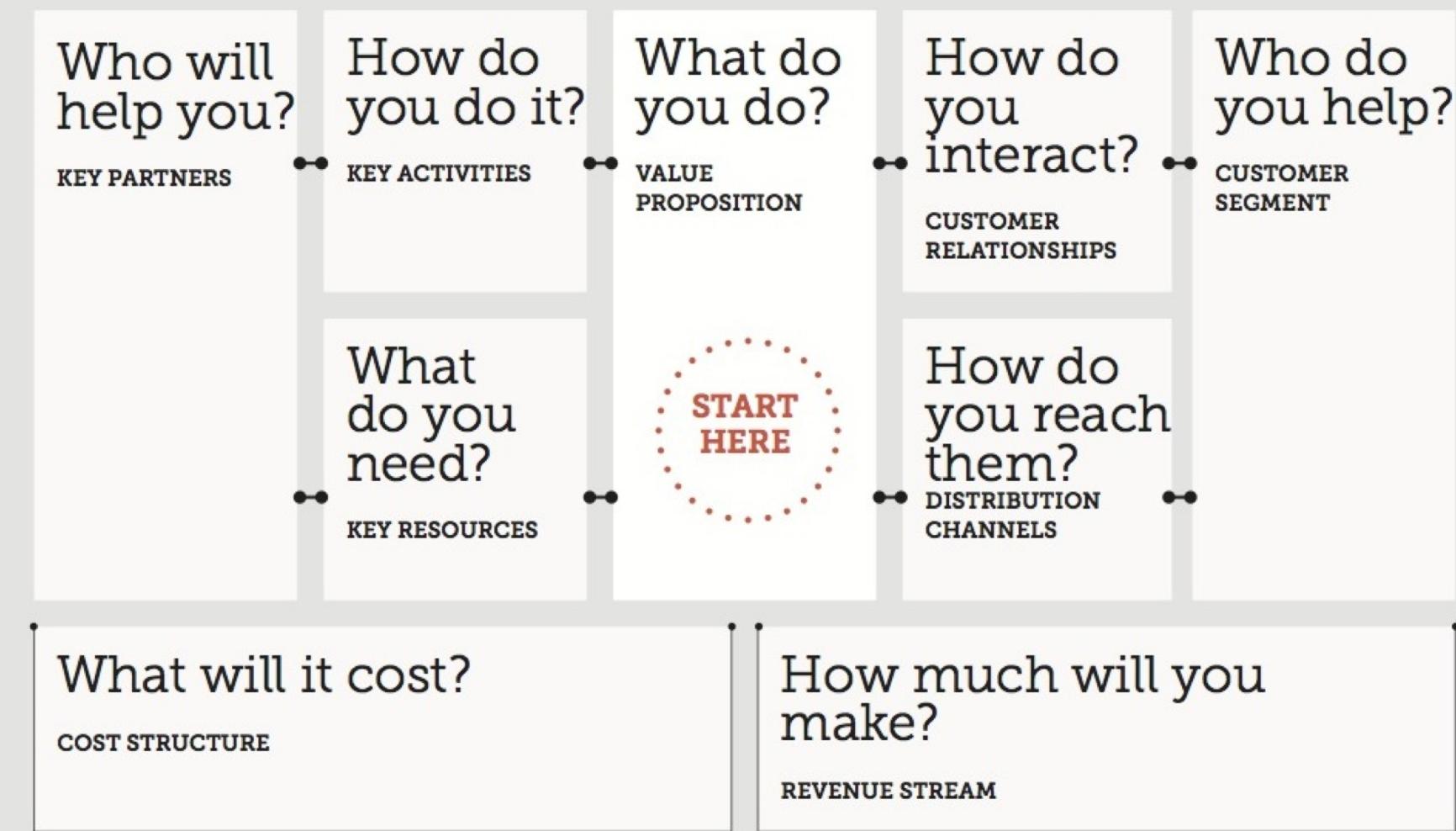








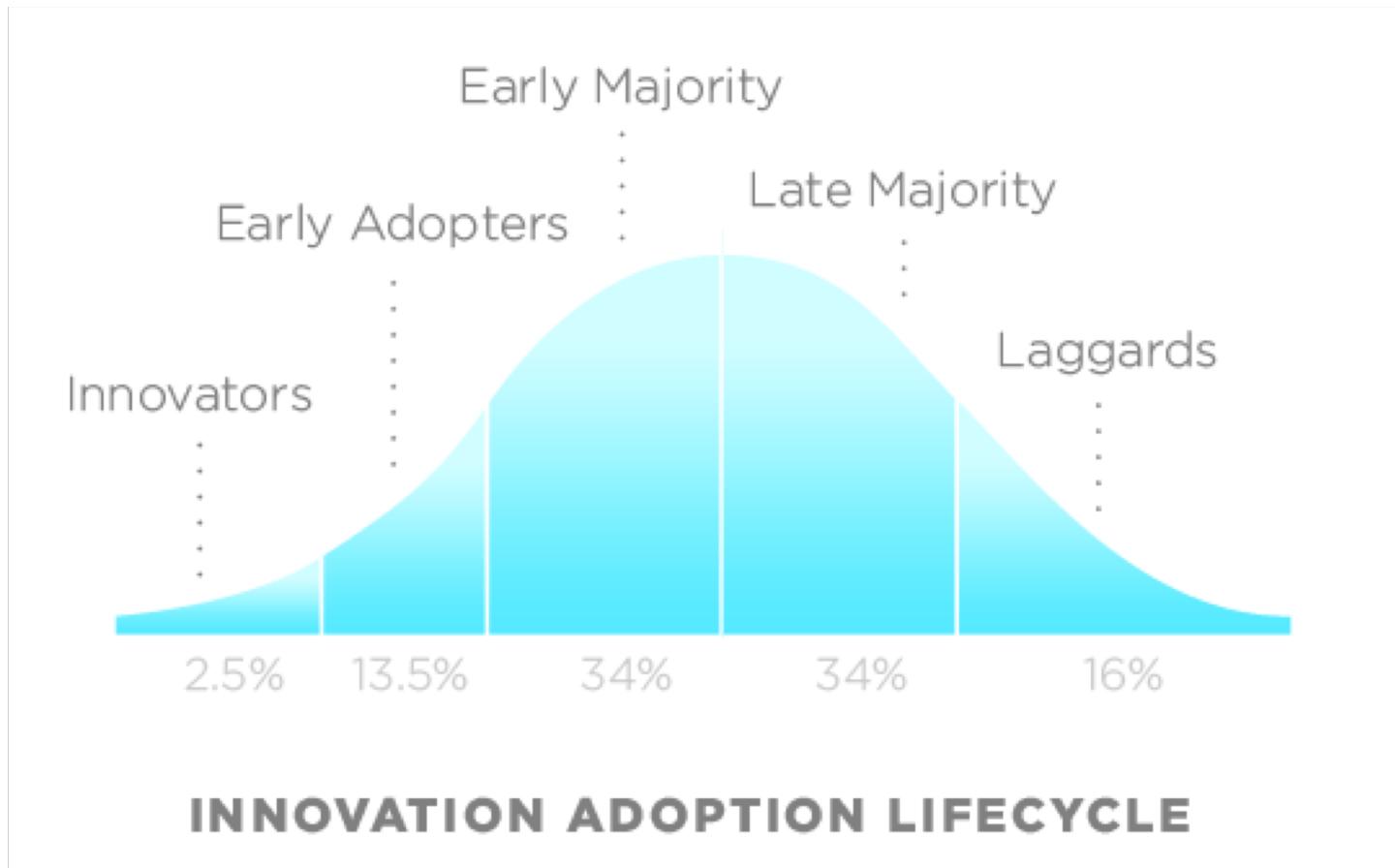
Business Model Canvas





Start

▪ Customer Segmentation





Value proposition

- Customers want not only the product but also a product which can **solve problem** and **add value**
 - What are the features of the core product?
 - What problems to solve?
 - Reducing time, convenience, low risk
 - What is the add value?
 - High quality, mulit-functions



Customer Segment

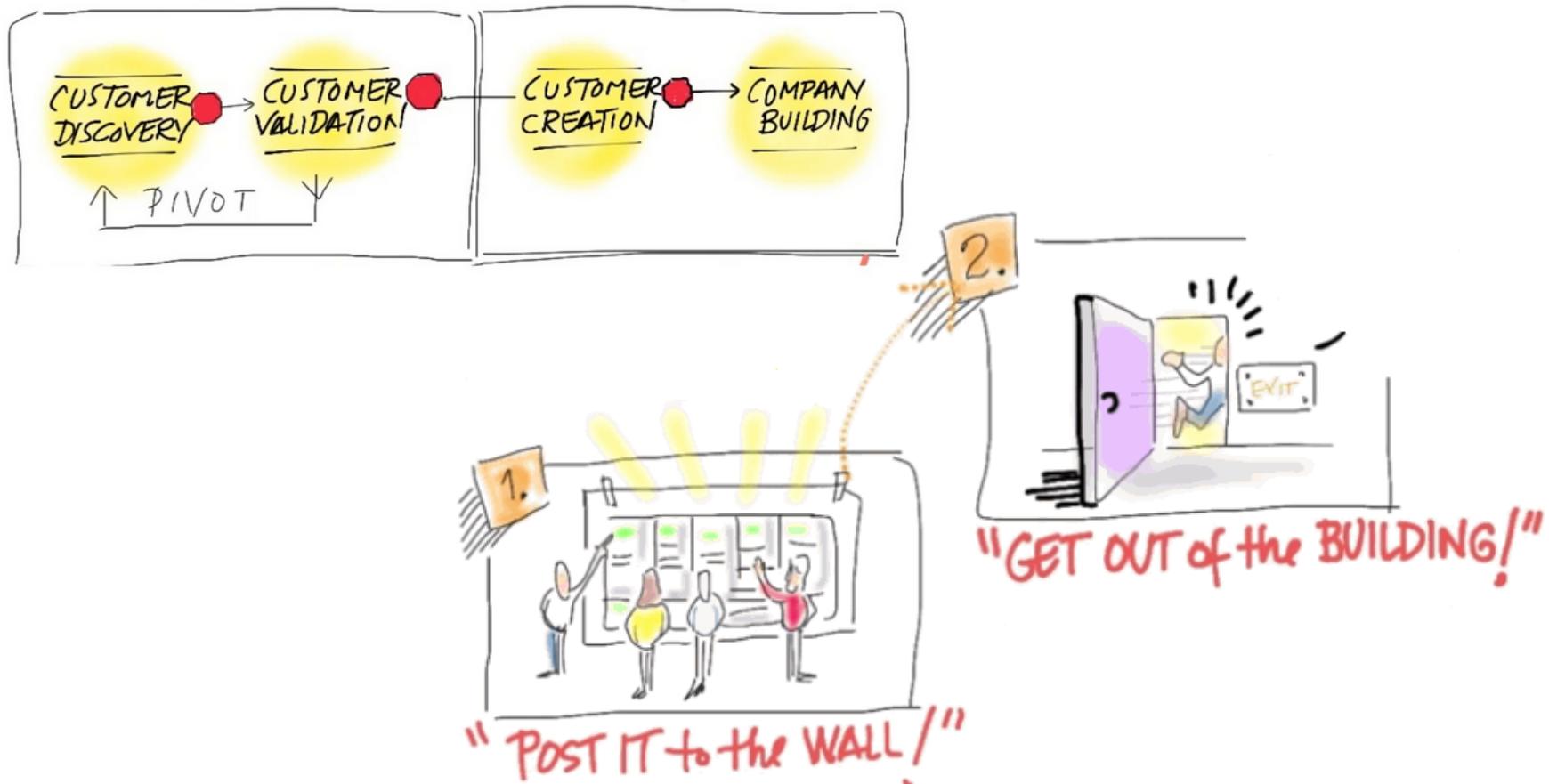
- Who are your customers?
 - Age, gender, location
- What are their usual requirements?
 - Working requirements, social, personal
- What are their problems?



Customer Developing



- From assumption to validation





Customer Discovery

- A. Questionnaire
- B. Focusing Group
- C. Interview (F2F)
- D. All of above





Customer Interview

- 3:1
- Customer behavior, Feedback
- Avoiding Confirmation Bias
- Straight forward
- 5 Whys
- Listening
- Recommend other customer



How to find the root of the problem



用户无法使用已购买的团购

为什么 → 用户无法预约

为什么 → 用户找不到商户电话

为什么 → 团购认领短信内没有电话号码

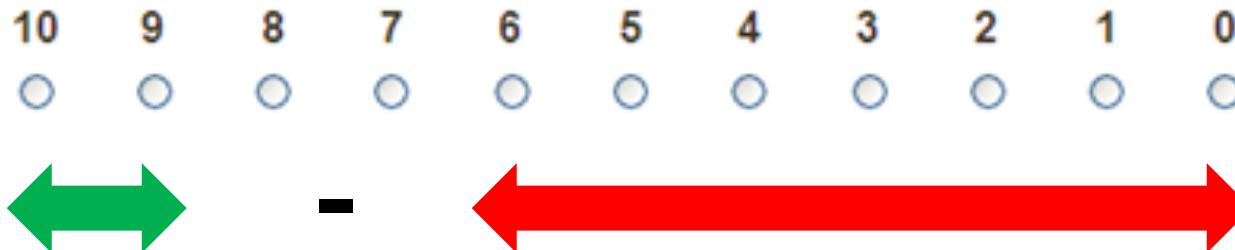
为什么 → POI没有电话信息

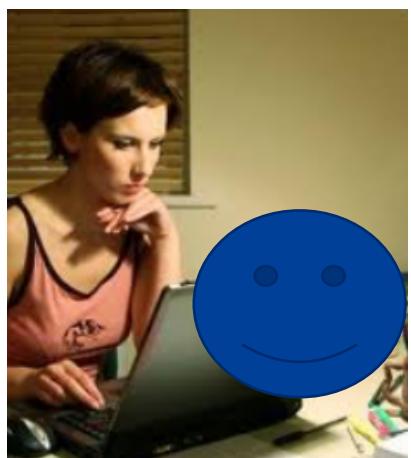


Questionnaire



- Quantizing customers requirements
- Get the NPS of product
 - NPS (Net Promoter Score)
 - How likely is it that you would recommend our company/product/service to a friend or colleague?





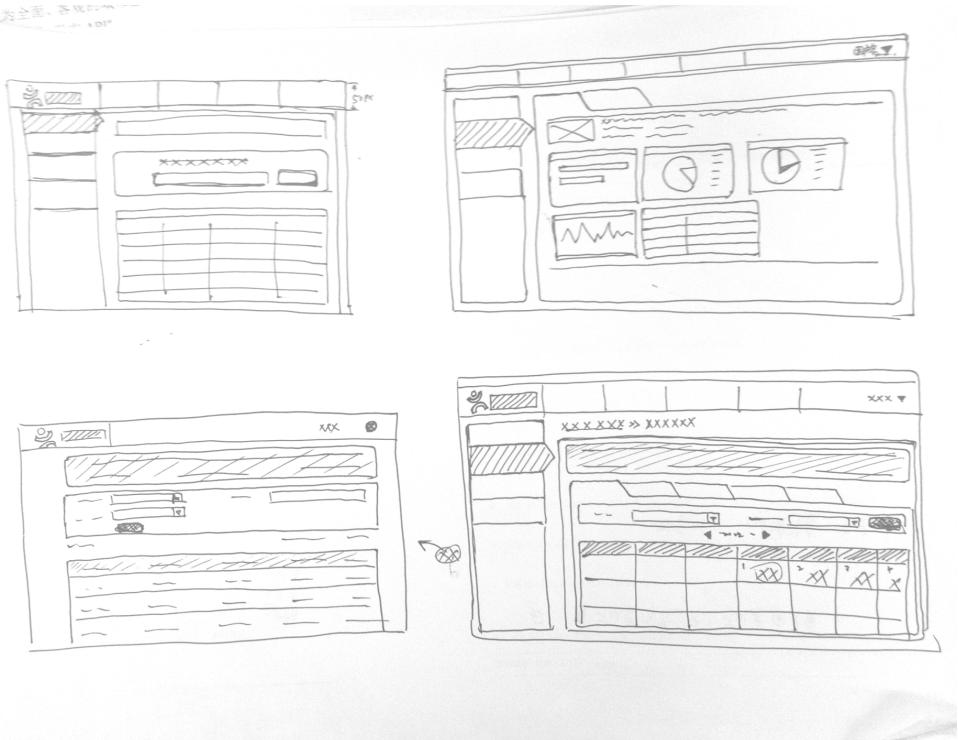
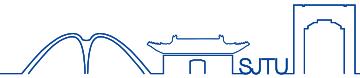


How to find the solution

- Existing solutions
- Experts & Consultants
- Brainstorming
- Product sense
- From Assumption to Validation



How to present?



Axure RP



Balsamiq Mockups



Customer Discovery



- Reducing the cost of learning
- Speeding up learning progress



MVP

- Minimum viable product as the smallest possible product that has three critical characteristics: people choose to use it or buy it; people can figure out how to use it; and we can deliver it when we need it with the resources available



Minimum



Too simple

MVP



Viable



Much



How to produce MVP



▪ Investigation, Validation, Abstraction

- 非餐饮行业：KTV，美容
- 人工客服电话
- 代替纸笔，做TMS系统
- 推PAD功能，短信，PC
- 做7*24小时实时预定
- 选座位，类似在线买电影票
- 预定优惠，提供积分鼓励
- 餐饮分类Web+App
- Message 类型丰富
- 批量提交预定N选 方式
- 商户结算是不做在线支付

- 包房预付费
- 预付押金点菜
- 自动配菜
- 提前排队领号
- CRM服务
- 开放平台
- 外卖服务
- Remarketing 营销服务
- 买断部分热门餐馆位置
- 支持团购业务
- 实时优惠信息发布
- Paid 优惠券



Software Innovation and R&D Management

CI/CD

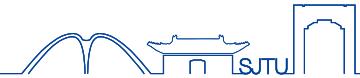
2019-03-13



上海交通大学
SHANGHAI JIAO TONG UNIVERSITY



Cloud Native



1. CONTAINERIZATION

- Commonly done with Docker containers
- Any size application and dependencies (even PDP-11 code running on an emulator) can be containerized
- Over time, you should aspire towards splitting suitable applications and writing future functionality as microservices

2. CI/CD

- Setup Continuous Integration/Continuous Delivery (CI/CD) so that changes to your source code automatically result in a new container being built, tested, and deployed to staging and eventually, perhaps, to production
- Setup automated rollouts, roll backs and testing



Cloud Native



3. ORCHESTRATION & APPLICATION DEFINITION

- Kubernetes is the market-leading orchestration solution
- You should select a Certified Kubernetes Distribution, Hosted Platform, or Installer: cncf.io/ck
- Helm Charts help you define, install, and upgrade even the most complex Kubernetes application





Cloud Native



4. OBSERVABILITY & ANALYSIS

- Pick solutions for monitoring, logging and tracing
- Consider CNCF projects Prometheus for monitoring, Fluentd for logging and Jaeger for Tracing
- For tracing, look for an OpenTracing-compatible implementation like Jaeger

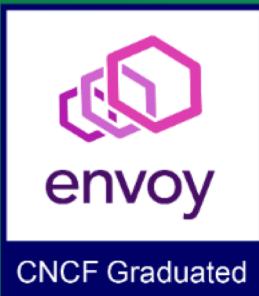




Cloud Native

5. SERVICE PROXY, DISCOVERY, & MESH

- CoreDNS is a fast and flexible tool that is useful for service discovery
- Envoy and Linkerd each enable service mesh architectures
- They offer health checking, routing, and load balancing





Cloud Native



7. DISTRIBUTED DATABASE & STORAGE

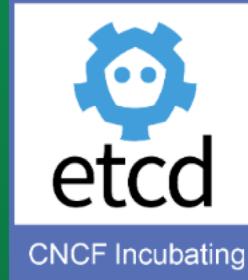
When you need more resiliency and scalability than you can get from a single database, Vitess is a good option for running MySQL at scale through sharding. Rook is a storage orchestrator that integrates a diverse set of storage solutions into Kubernetes. Serving as the "brain" of Kubernetes, etcd provides a reliable way to store data across a cluster of machines.



CNCF Incubating



CNCF Incubating



CNCF Incubating

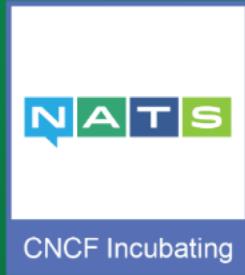


Cloud Native



8. STREAMING & MESSAGING

When you need higher performance than JSON-REST, consider using gRPC or NATS. gRPC is a universal RPC framework. NATS is a multi-modal messaging system that includes request/reply, pub/sub and load balanced queues.



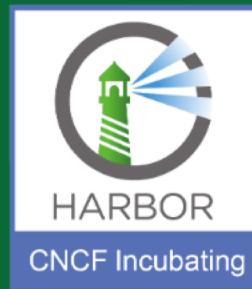


Cloud Native



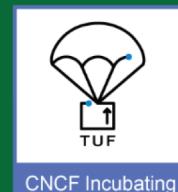
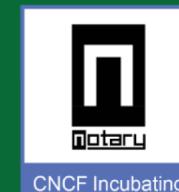
9. CONTAINER REGISTRY & RUNTIME

Harbor is a registry that stores, signs, and scans content. You can use alternative container runtimes. The most common, all of which are OCI-compliant, are containerd, rkt and CRI-O.



10. SOFTWARE DISTRIBUTION

If you need to do secure software distribution, evaluate Notary, an implementation of The Update Framework.

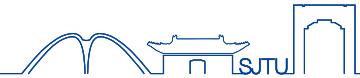


Introduction



- Continuous integration
 - The practice of merging all developer working copies to a shared mainline several times a day
 - “The practice where members of team integrate their work frequently; usually each person integrates at least daily – leading to multiple integrations per day.” – Martin Fowler

Introduction



- Continuous Delivery
 - “A software discipline where you build software in such a way that software can be released to production at any time.” – Martin Fowler
 - A software engineering approach in which teams produce software in short cycles, ensuring that the software can be reliably released at any time.



Introduction

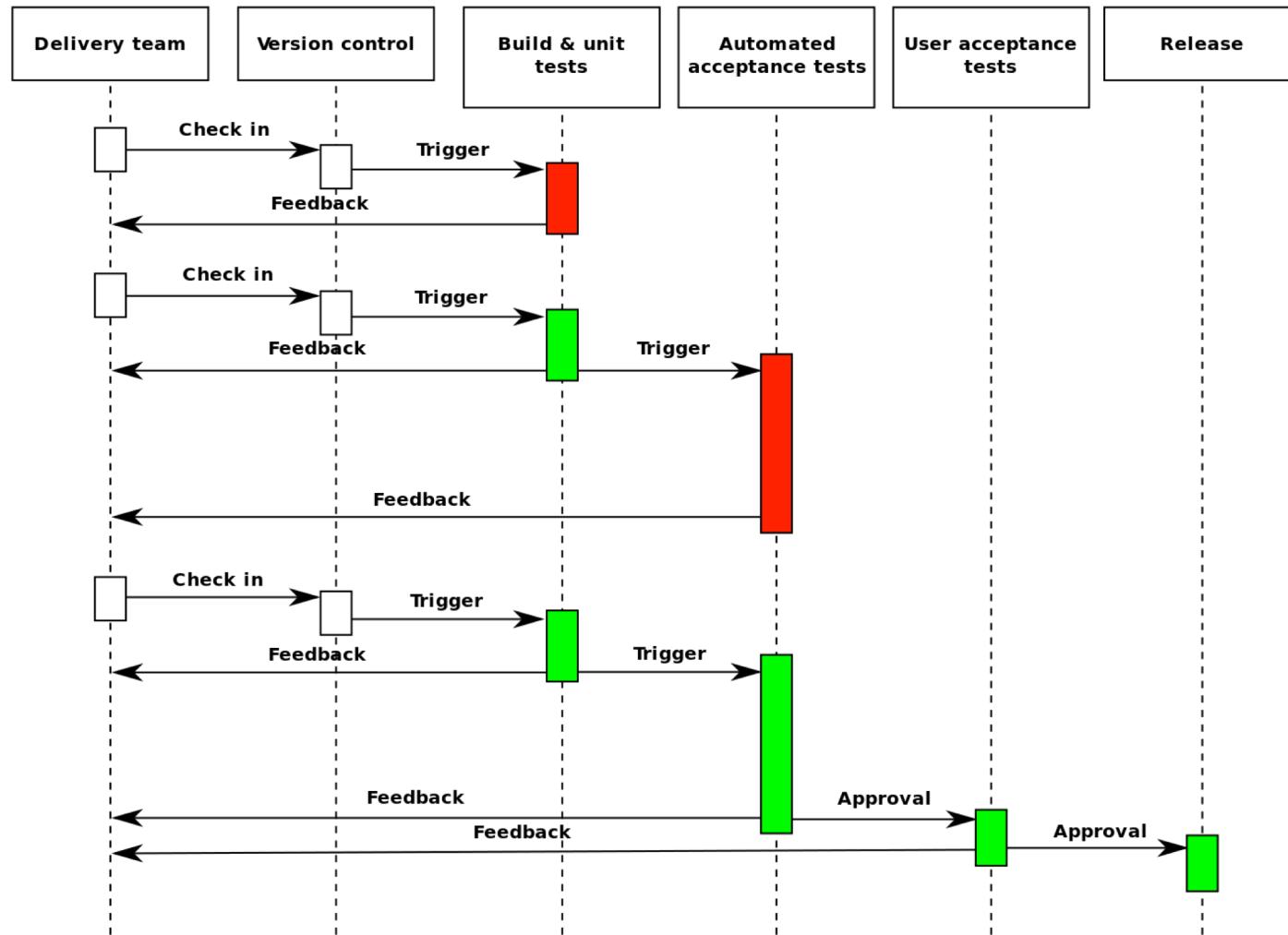


▪ Continuous Deployment

- “Every changes you make automatically get deployed through the deployment pipeline.” – Martin Fowler
- A software engineering approach in which teams produce software in short cycles, ensuring that the software can be reliably released at any time.



Introduction





CI

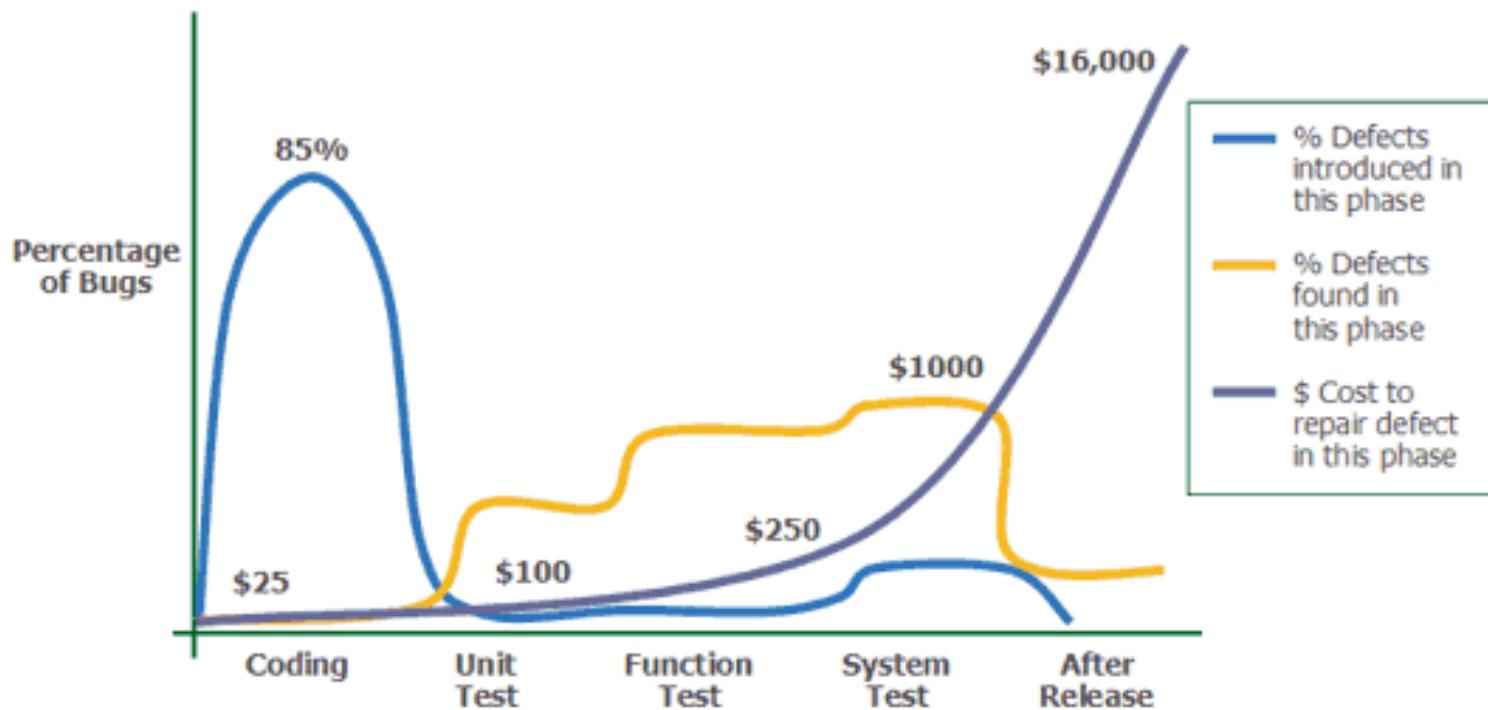
- How to build software *better, faster, cheaper*
 - A development methodology
 - Daily developer integrations
 - Verified by automated builds

Continuous Compilation
!=
Continuous Integration



CI

- Why do it?
 - Fix bugs lately



CI



- Why do it?
 - Team cohesion
 - The quality of team code base
 - Project visibility
 - Deployable software



CI - Better



- Build better quality software
- That is tested early and often
- That adheres to best practices and coding standards



CI - Faster



- Test in parallel, not at the end
- No integration points
- Builds become a non event



CI - Cheaper

- Identify defects earlier
- Fix when least costly
- Easily repeatable testing



CI - Build

- Compilation
 - Ensures code actually compiles
 - On every target platform
- Test Execution
 - Ensures product functions as expected
 - Through repeatable testing
- Database Integration
 - Ensures DB and code in sync
 - Automates (re)creation of DB and test data



CI - Build



- Code Inspection
 - Ensures a healthy code base
 - Identifies problems early
 - Enforces best practices
- Automated Deployment
 - Products can be released anytime
 - Continually demo-able state
 - Eliminates “works on my machine”
- Documentation Generation
 - Ensures documentation is current
 - Removes burden from developers
 - Produces build reports and metrics



CI – Getting started

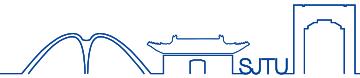
- When to build
 - At every check-in
 - Every time a dependency changes
- How to build
 - Use a single build script
 - That can run from the cmdline
 - Do not depend on an IDE
- How to be continuous
 - Use a dedicated CI server, not cron
 - Trigger on all daily check-ins
 - Not [only] at midnight



CI – Best Practice

- Commit Early, Commit Often
- Never commit broken code
- Fix build failures immediately
- Fail Fast
- Act on metrics
- Build in every target environment
- Create artifacts from every build

CI - Summary



- Help you build **better quality** software **faster**, with more **confidence**



Best Practice



- Dev Env
 - IDE
- Code Repository
 - GitHub
 - Gitlab
 - BitBucket
- CI
 - Jenkins
 - Travis CI
 - drone
- CD
 - Heroku
 - GCE
 - AWS



Best Practice

- Container
 - Docker, lxc, rkt, runC, Pouch
- Orchestrator
 - Kubernetes
- Monitor
 - CAT



About the developer

- Architect that doesn't understand **technology** will leads to the wrong solution
- Architect that doesn't understand the **business** leads to wrong problem and no solution
- Developers that doesn't understand the **business** leads to wrong solution to every problem
- Developer not understanding **technology** leads to no solution for no problems.