

# INFO5992 EXAM CHEATSHEET

Understanding IT Innovations - Complete Reference

## 1. Innovation Types & Strategies

### 1.1 Disruptive Innovation

- **Low-End Disruption:** Targets overserved customers with “good enough” cheaper solutions → incumbents ignore
- **New-Market Disruption:** Creates new market serving non-consumers → different value proposition
- **Sustaining Innovation:** Improves existing products along established dimensions → incumbents excel

### 1.2 Innovation Dilemma

- Incumbents allocate resources to high-margin customers
- Ignore low-end threats until too late
- Existing capabilities become obsolete

### 1.3 Competence Types

- **Competence-Enhancing:** Builds on existing knowledge → incumbents have advantage
- **Competence-Destroying:** Requires new capabilities → startups have advantage (clean slate)

### 1.4 Technological Discontinuity

**Definition:** Dramatic breakthrough that makes existing tech obsolete

#### Why Companies Choose It:

- Overcome competitive stagnation
- Access new markets/capabilities
- Leapfrog competitors
- Respond to market shifts

#### Types:

- *Competence-enhancing:* Leverages existing skills
- *Competence-destroying:* Requires new expertise

#### Impact on Startups:

- Opportunity window for new entrants
- Incumbents' advantages neutralized
- Market uncertainty → experimentation phase

## 2. Dominant Design & Lifecycle

### 2.1 Dominant Design Phases

1. **Fluid Phase:** Multiple competing designs, high uncertainty, product performance focus
2. **Transitional Phase:** Design standards emerge, competition intensifies
3. **Specific Phase:** One design dominates, shift to process innovation & cost reduction

### 2.2 Why Dominant Design Matters

- Reduces uncertainty for customers & producers

- Enables economies of scale & network effects
- Compatibility with ecosystem/infrastructure
- Signals stability to mainstream market

### 2.3 When Dominant Design May NOT Emerge

- High customization needs (different segments need different designs)
- Rapid technological change prevents standardization
- Strong network effects lock in multiple competing standards
- Regulatory fragmentation across markets
- Low switching costs → continuous experimentation

### 2.4 Choosing Dominant Design (Startup)

- Customer acceptance & reduced adoption risk
- Compatibility with existing systems
- Cost & scalability for mass market
- Performance reliability benchmarks
- Network effects & ecosystem leverage

## 3. Technology Adoption Lifecycle

### 3.1 Adoption Segments

1. **Innovators** (2.5%): Tech enthusiasts, high risk tolerance
2. **Early Adopters** (13.5%): Visionaries, willing to experiment
3. **Early Majority** (34%): Pragmatists, need proven solutions
4. **Late Majority** (34%): Conservatives, adopt when necessary
5. **Laggards** (16%): Skeptics, resist change

### 3.2 Crossing the Chasm

- **The Gap:** Between Early Adopters & Early Majority
- **Why Hard:** Early majority demands reliability, references, established standards; startups lack credibility
- **How Dominant Design Helps:** Reduces uncertainty, signals stability, builds trust, enables ecosystem

## 4. Technology Hype Cycle

### 4.1 5 Stages

1. **Innovation Trigger:** Breakthrough sparks interest
2. **Peak of Inflated Expectations:** Unrealistic hype, many experiments
3. **Trough of Disillusionment:** Reality fails expectations, failures occur
4. **Slope of Enlightenment:** Practical applications emerge, *dominant design often emerges here*
5. **Plateau of Productivity:** Mainstream adoption, standards solidify

**Relationship with Dominant Design:** Competing designs

in early stages → dominant design stabilizes market during Slope/Plateau → reduces uncertainty → enables scale

## 5. Value Chain vs Value Network

### 5.1 Value Chain (Porter)

- **Linear model:** Sequential activities adding value
- **Primary activities:** Inbound logistics, operations, outbound logistics, marketing/sales, service
- **Support activities:** Infrastructure, HR, tech development, procurement
- **Focus:** Internal efficiency, cost reduction
- **Example:** Manufacturing (raw materials → production → distribution → retail)

### 5.2 Value Network

- **Network model:** Multiple players co-create value
- Interdependent relationships, not linear
- Value created through interactions & connections
- **Focus:** Ecosystem collaboration, network effects
- **Example:** Platform (app developers + users + payment providers all create value together)

### 5.3 Key Differences

Value Chain	Value Network
Linear flow	Multi-directional
Single firm focus	Ecosystem focus
Sequential activities	Simultaneous interactions
Efficiency-driven	Network effects-driven

## 6. Value Proposition Canvas

### 6.1 Customer Profile

- **Jobs:** Tasks customers accomplish (functional, social, emotional)
- **Pains:** Obstacles, risks, negative emotions
- **Gains:** Desired outcomes, benefits, aspirations

### 6.2 Value Map

- **Products/Services:** What you offer
- **Pain Relievers:** How you eliminate/reduce pains
- **Gain Creators:** How you create benefits

### 6.3 Achieving Fit

- Align value map with customer profile
- Address important jobs, alleviate major pains, create meaningful gains
- = **Product-Market Fit (PMF)**

### 6.4 Value Types

- **Functional:** Solves practical problem
- **Emotional:** Makes customer feel better

- **Social:** Status, belonging, impression
- **Life-Changing:** Transforms customer's life

## 7. Customer Development Process

### 7.1 4 Stages

1. **Customer Discovery:** Test problem hypotheses, identify real pains/needs, get out of building
2. **Customer Validation:** Test solution with MVP, achieve repeatable sales model
3. **Customer Creation:** Scale demand, execute marketing/sales
4. **Company Building:** Transition to execution-focused organization

### 7.2 Key Principles

- **Get Out of Building:** Talk to real customers, not desk research
- **Build-Measure-Learn:** Fast iteration cycles
- **Pivot or Persevere:** Change strategy based on validated learning
- **MVP:** Test assumptions with least effort

### 7.3 Creating Value in Customer Dev

- Align value map with customer profile through interviews
- Validate problem-solution fit before building
- Iterate based on feedback to strengthen value proposition
- Test with prototypes/wireframes in discovery phase
- Build high-fidelity MVP in validation phase
- Verify scalability & repeatability of business model

## 8. Lean Startup vs Waterfall

### 8.1 Why Waterfall Fails for Startups

- Linear & rigid → can't adapt to feedback
- Late customer involvement → build wrong product
- High waste if assumptions wrong
- Assumes clear requirements upfront (unrealistic for innovation)

### 8.2 Lean Advantages

- **Build-Measure-Learn loop:** Fast prototyping & testing
- Early customer involvement → validate assumptions
- Flexible pivots based on feedback
- Resource-efficient → focus on essential features
- Reduces risk through validated learning
- Supports experimentation & iteration

## 9. Open Innovation

### 9.1 3 Innovation Flows

1. **Outside-In:** Bring external ideas/tech into company (licensing in, partnerships, crowdsourcing)
2. **Inside-Out:** Commercialize internal ideas externally (spin-offs, licensing out, selling IP)
3. **Coupled:** Combine both through alliances, joint ventures, co-creation

### 9.2 Open Source vs Proprietary

Open Source	Proprietary
Community-driven innovation	Controlled development
Lower costs (no licensing)	Revenue from licenses
Transparency & auditability	Trade secrets protected
Faster iteration	Competitive advantage
Interoperability & standards	Lock-in strategy

### 9.3 Why Open Source Drives Innovation (4+ Reasons)

1. **Faster innovation:** Global community contributes improvements continuously
2. **Lower costs:** No licensing fees → allocate resources to differentiation
3. **Transparency:** Code auditability builds trust (critical for finance, healthcare)
4. **Interoperability:** Easier integration with existing systems via standards
5. **Flexibility:** Customize to specific needs without vendor lock-in
6. **Community support:** Faster bug fixes, security patches, feature development
7. **Talent attraction:** Developers prefer working with open technologies

### 9.4 Open Source Benefits for Startups

- **Standardization:** Compliance with industry norms (e.g., Linux, PostgreSQL)
- **Interoperability:** Easy integration with existing systems
- **Community innovation:** Continuous improvements (security, performance)
- **Lower costs:** Allocate resources to core differentiation
- **Trust:** Code auditability for regulatory compliance

### 9.5 Closed to Open Innovation Shift

Reasons for Shift:

- Access external expertise & reduce R&D costs
- Faster time-to-market through collaboration
- Tap into global talent pool
- Share development risk with ecosystem

### Risks:

- Loss of IP control & competitive advantage
- Quality control challenges with external contributors
- Coordination overhead & integration complexity
- Risk of free-riding by competitors

## 10. APIs & Modularity

### 10.1 API Types

1. **API as Product:** Core business offering (Stripe, Twilio, AWS)
2. **API Enhancing:** Adds functionality to existing products (Google Maps API)
3. **API Promoting:** Drives adoption & ecosystem growth (Twitter API)

### 10.2 How APIs Enable Innovation

- **Modularity:** Decouple components → independent development
- **Scalability:** Add features without rebuilding core system
- **Third-party integration:** External developers extend functionality
- **Ecosystem growth:** Network effects as more developers build on platform
- **Faster iteration:** Update modules independently

### 10.3 Modularity in Software Architecture

- **Definition:** Breaking system into independent, interchangeable components
- **Benefits:**
  - Parallel development by different teams
  - Easier testing & debugging
  - Component reusability across projects
  - Lower coupling → changes don't cascade
  - Enable third-party innovation via APIs
- **Example:** Microservices architecture (authentication, payment, notification as separate modules)

## 11. Platform Business Models

### 11.1 Platform Ecosystem Model

- **Integrator Platform:** Mediates between external innovators & customers
- Controls transactions & interactions
- Benefits from network effects
- Monetizes through fees, subscriptions, data

## 11.2 Key Players in Platform

- **Producers:** Create value (app developers, content creators)
- **Consumers:** Use value (end users)
- **Platform Owner:** Provides infrastructure & governance
- **Complementors:** Enhance platform value (device makers, payment providers)
- **Regulators:** Ensure compliance & safety

## 11.3 Platform Types

1. **Transaction platforms:** Facilitate exchanges (Uber, Airbnb, eBay)
2. **Innovation platforms:** Enable third-party development (iOS, Android, AWS)
3. **Integrated platforms:** Combine both (Apple ecosystem)

## 11.4 Platform Monetization

- Transaction fees (commission on each sale)
- Subscription fees (premium features)
- Advertising revenue (targeted ads)
- Freemium model (free basic, paid premium)
- Data monetization (insights, analytics)

## 12. Crowdsourcing

### 12.1 4 Types

1. **Knowledge Discovery:** Tap distributed expertise (Innovenative - scientific problems)
2. **Broadcast Search:** Post problems, best solution wins (Kaggle - data science competitions)
3. **Peer-Vetted Creative:** Community evaluates ideas (Threadless - t-shirt designs)
4. **Distributed Human Intelligence Tasks:** Micro-tasks at scale (Amazon MTurk - labeling, surveys)

### 12.2 Challenges

- Quality control & reliability of contributions
- Intellectual property ownership disputes
- Participant motivation & retention
- Coordination costs & management overhead
- Free-riding & unequal contribution

### 12.3 Benefits

- Access to diverse expertise globally
- Cost-effective compared to in-house R&D
- Faster problem-solving through parallel efforts
- Identify innovative solutions from unexpected sources

## 13. Lead Users

### 13.1 Who They Are

- Experience needs ahead of market (future-oriented)
- High benefit from solutions (strong motivation)

- Often innovate solutions themselves (proactive)

## 13.2 Why Valuable for Sustainable Innovation

- Provide insights into future market needs
- Co-create products & provide early feedback
- Accelerate adoption as opinion leaders
- Reduce uncertainty & validate design for scalability
- Test extreme use cases that normal users won't encounter

## 13.3 Why Normal Users Less Effective (5 Reasons)

1. **Lack vision for future needs:** Focus on immediate problems, not emerging trends
2. **Limited technical expertise:** Can't suggest advanced/disruptive solutions → incremental feedback only
3. **Lower risk appetite:** Prefer proven solutions, avoid untested tech → slow radical innovation
4. **Incremental feedback only:** Minor improvements, not breakthrough ideas
5. **Biased by dominant designs:** Accustomed to current standards → resist creative thinking

## 14. Knowledge Sharing in Startups

### 14.1 How It Drives Innovation

- **Cross-pollination of ideas:** Different perspectives spark creativity
- **Faster problem-solving:** Collective intelligence reduces bottlenecks
- **Reduced duplication:** Share learnings to avoid repeated mistakes
- **Skill development:** Team learns from each other
- **Organizational learning:** Capture & reuse tacit knowledge

### 14.2 Mechanisms/Practices for Knowledge Sharing

1. **Cross-functional teams:** Break silos, share domain expertise
2. **Regular knowledge-sharing sessions:** Sprint demos, brown bags, tech talks
3. **Documentation culture:** Wikis, runbooks, design docs
4. **Mentorship & pair programming:** Tacit knowledge transfer
5. **Retrospectives:** Capture lessons learned from projects
6. **Internal communities of practice:** Groups focused on specific domains
7. **Open communication tools:** Slack channels, forums for async sharing

## 15. Coopetition

### 15.1 Definition

Collaboration with competitors instead of pure disruption

## 15.2 Why Startups Choose It

- **High entry barriers:** Costly infrastructure, regulations → partnerships needed
- **Shared resources:** Access to distribution, tech, customer base
- **Network effects:** Faster adoption & ecosystem growth through collaboration
- **Risk reduction:** Lower financial & market uncertainty
- **Complementary strengths:** Startup innovation + incumbent scale/credibility

## 15.3 Examples

- Fintech startups partnering with banks (access to customers + regulatory compliance)
- EV startups using established automakers' charging networks
- SaaS companies integrating with competitors to serve customers better

## 16. Organizational Structures

### 16.1 Mechanistic Structure

- High formalization, centralized decisions
- Rigid hierarchy, clearly defined roles
- Efficient for stability, routine tasks
- Limits creativity & flexibility

### 16.2 Organic Structure

- Low formalization, decentralized decisions
- Flexible processes, collaborative
- Encourages adaptability & innovation

### 16.3 Making Mechanistic Innovative

1. **Cross-functional teams:** Encourage collaboration across departments
2. **Reduce formalization:** Allow flexibility in processes
3. **Promote risk-taking:** Failure viewed as learning, not punishment
4. **Allocate innovation time:** e.g., Google's "20% time"
5. **Reward creativity:** Incentives for innovative ideas
6. **Skunk Works teams:** Small, autonomous groups for disruptive projects
7. **Flatten hierarchy:** Reduce approval layers for faster decisions

## 17. Startup Failure Factors

### 17.1 Top 3 Reasons

1. **Lack of Product-Market Fit:** Product doesn't solve real customer needs → poor adoption. Built without validating demand.

- Insufficient Capital:** Run out of cash before achieving scale. High burn rates, long R&D cycles (hardware/deep-tech).
- Weak Business Model:** No clear revenue model, mispricing, unclear monetization → unsustainable operations.

## 17.2 Other Contributing Factors

- Poor team dynamics & founder conflicts
- Inability to pivot when assumptions fail
- Intense competitive pressure
- Premature scaling before PMF
- Ignoring customer feedback

## 18 18. Business Model Canvas

### 18.1 Key Components

**Customer Segments:** Who are you serving?

- Mass market, niche, segmented, diversified, multi-sided

**Value Propositions:** What value do you deliver?

- Newness, performance, customization, design, brand, price, convenience, risk reduction

**Key Resources:** What assets required?

- Tangible:** Physical (facilities, equipment), Financial (cash, credit)
- Intangible:** Intellectual (patents, IP, data), Human (skills, expertise, talent)

**Key Activities:** What do you do?

- Production, problem-solving, platform/network management

**Revenue Streams:** How do you make money?

- Asset sale, usage fee, subscription, licensing, advertising, freemium

### 18.2 Why Customer Segments & Value Props Critical

- Focus resources:** Target right customers, avoid wasting effort
- Achieve PMF:** Clear value prop aligned with segment needs
- Differentiation:** Stand out from competitors in specific segments
- Scalability:** Repeatable model for similar customers
- Revenue model:** Pricing & monetization fit customer willingness to pay

## 19 19. Diffusion of Innovation (Rogers, 2003)

### 19.1 Definition

Diffusion is the process by which an innovation is communicated through certain channels over time among members of a social system.

### 19.2 Rogers' 5 Key Elements

- Innovation:** Product, process, or business model innovation
- Communication Channels:** Word-of-mouth, TV, Internet, Social Media
- Time:** Rate of adoption
- Members:** Different adoption timing (Innovators → Laggards)
- Social System:** External (media, govt) & internal (opinion leaders) influencers

### 19.3 Innovation-Development Process

- Recognizing Problem/Need
- Research
- Development
- Commercialization
- Diffusion/Adoption
- Consequences

### 19.4 Market Entry Strategies

Strategy	Failure Rate	Description
First Mover	47%	Enter quickly, hope to dominate
Second Mover	8%	"Me-too" product, low cost
Fast Second	8%	Enter as design EMERGES

**Fast Second** is optimal! Examples: Microsoft (IE, Xbox), Amazon, Canon, TikTok, Lyft

### 19.5 Factors Leading to Dominant Design

- Learning Effects:**
  - Greater knowledge → Greater use → Faster improvement
  - Cumulative learning drives quality/efficiency gains
- Network Effects:**
  - Direct:** Value ↑ with more users (e.g., Telephone)
  - Indirect:** Complementary goods ↑ (e.g., PC Architecture)
  - Two-sided:** Two user groups benefit (e.g., eBay)
  - Local:** Network benefits in groups (e.g., IM)
- Government Regulation:**
  - Mandates common design (e.g., Digital TV standards)
- Diffusion Strategies for Tech Startups**
  - Target early adopters:** Identify visionaries willing to take risks, use as reference customers
  - Build credibility:** Case studies, testimonials, proof of concept with reputable customers
  - Reduce adoption barriers:** Free trials, freemium model,

easy onboarding

- Leverage network effects:** Incentivize referrals, viral loops, community building
- Partner with complementors:** Integrate with established platforms/ecosystems
- Educate market:** Content marketing, webinars, thought leadership to build awareness
- Align with dominant design:** Reduce perceived risk for mainstream adopters

### 19.7 Barriers to Innovation Implementation

- Organizational inertia:** Resistance to change, "not invented here" syndrome, fear of disrupting existing business
- Resource constraints:** Insufficient budget, talent shortage, competing priorities
- Lack of leadership support:** Innovation not prioritized, risk-averse culture
- Poor communication:** Silos prevent knowledge sharing, misaligned incentives

## 20 20. Agentic AI vs Autonomous

### 20.1 Agentic AI

- Proactive & goal-driven:** Pursues objectives independently
- Adaptive reasoning:** Adjusts strategy based on environment
- Multi-step planning:** Breaks down complex tasks
- Example:** AI research assistant formulating hypotheses & designing experiments

### 20.2 Autonomous Systems

- Reactive & task-specific:** Executes predefined tasks
- Rule-based:** Follows programmed instructions
- Limited adaptability:** Within narrow scope
- Example:** Self-driving car following traffic rules

## 21 21. General Purpose Technology

### 21.1 Characteristics of GPT

- Wide applicability across sectors
- Continuous improvement over time
- Spawns complementary innovations
- Transforms entire economies
- Examples: Electricity, internet, steam engine

### 21.2 Can Generative AI be GPT?

YES - Supporting Arguments:

- Wide applicability:** Content creation, coding, design, research, customer service across all industries
- Continuous improvement:** Rapid iteration, models im-

proving exponentially

3. **Complementary innovations:** New tools, applications, business models emerging

4. **Productivity transformation:** Automating knowledge work at scale

**Counterarguments (if needed):**

- Still early stage, adoption not yet universal
- Economic transformation impact unclear

## 22. EV Types (if relevant)

### 22.1 Battery Electric Vehicle (BEV)

- 100% electric, no combustion engine
- Zero tailpipe emissions
- **Example:** Tesla Model 3, Nissan Leaf

### 22.2 Plug-in Hybrid (PHEV)

- Electric motor + combustion engine
- Can run on electric only (limited range), then switches to gas
- **Example:** Toyota Prius Prime, Chevrolet Volt

### 22.3 Range-Extended EV (REEV)

- Primarily electric, small engine as generator (doesn't drive wheels)
- Extends range when battery depleted
- **Example:** BMW i3 with range extender

## 23. Exam Answer Templates

### 23.1 2-Mark Question (2-3 lines)

- Define concept in 1 line
- Key reason/application in 1-2 lines
- **Example:** “PMF is degree to which product satisfies market demand. Value Prop Canvas achieves it by aligning customer jobs/pains/gains with pain relievers & gain creators.”

### 23.2 5-Mark Question

- 2-3 key points with brief explanation
- Each point: 1-2 lines
- Use bullet format
- **Example:** See Section 1 questions in answer key

### 23.3 10-Mark Scenario

- Apply framework clearly

- 4-5 bullet points
- Each point: claim + reasoning + scenario application
- Use keywords marker looks for
- **Example:** See Section 2 & 3 questions in answer key

## 24. 24. Key Exam Keywords

### 24.1 Innovation Type

disruptive, sustaining, low-end, new-market, competence-destroying, competence-enhancing, radical, incremental

### 24.2 Adoption

chasm, early adopters, mainstream, pragmatists, proven solution, references, credibility

### 24.3 Design

dominant design, standardization, compatibility, network effects, economies of scale, interoperability

### 24.4 Customer Dev

MVP, pivot, validated learning, get out of building, problem-solution fit, PMF, iteration

### 24.5 Value Prop

jobs-pains-gains, pain relievers, gain creators, functional/emotional/social value, fit

### 24.6 Lean

Build-Measure-Learn, iterate, waste reduction, customer feedback loops, fast experimentation

### 24.7 Open Innovation

outside-in, inside-out, coupled, ecosystem, interoperability, collaboration

### 24.8 Platform

network effects, multi-sided, transaction fees, ecosystem, complementors

## 25. 25. Quick Reference Frameworks

### 25.1 When Analyzing Innovation Type

1. Does it target overserved customers with cheaper solution? → *Low-end disruption*
2. Does it create new market for non-consumers? → *New-market disruption*
3. Does it improve along existing dimensions? → *Sustaining*
4. Does it make existing capabilities obsolete? →

*Competence-destroying*

### 25.2 When Analyzing Platform Ecosystem

1. Identify key players (producers, consumers, complements)
2. Show mediation role of platform
3. Explain network effects
4. Discuss monetization strategy

### 25.3 When Analyzing Customer Development

1. Discovery: What problem? Get out of building, test hypotheses
2. Validation: Does solution work? MVP, test sell, positioning
3. Pivot considerations: What's wrong? How to address?
4. Use Build-Measure-Learn terminology

### 25.4 When Analyzing Value Proposition

1. Customer jobs (what trying to do?)
2. Pains (what frustrations?)
3. Gains (what benefits desired?)
4. How product addresses each

## 26. 26. Common Question Patterns

### 26.1 “What type of innovation is X?”

- State innovation type clearly
- Provide 2 reasons with explanation
- Reference scenario specifics

### 26.2 “How can X help achieve Y?”

- Briefly define X
- List mechanisms/ways (3-4 points)
- Connect to outcome Y

### 26.3 “Why do startups fail / choose competition / etc?”

- List 3-5 clear reasons
- Each reason: 1-2 line explanation
- Use scenario context if given

### 26.4 “Explain difference between X and Y”

- Define both briefly
- Highlight 3-4 key differences
- Use table if helpful
- Provide example for each

## 27 26A. Week 2 - Detailed Additions

### 27.1 Joy's Law (Bill Joy, Sun Microsystems, 1990)

"Most of the bright people don't work for you – no matter who you are. [So] you need a strategy that allows for innovation occurring elsewhere."

### 27.2 EV Types

- **BEV**: Battery Electric Vehicle (fully electric)
- **PHEV**: Plug-in Hybrid Electric Vehicle
- **REEV**: Range-Extended Electric Vehicle
- **FCEV**: Fuel Cell Electric Vehicle

## 28 26B. Week 5 - FOSS Detailed

### 28.1 8-Step Crowdsourcing Process

1. Company has a problem
2. Company broadcasts problem online
3. Online "crowd" is asked to give solutions
4. Crowd submits solutions
5. Crowd vets solutions
6. Company rewards winning solvers
7. Company owns winning solutions
8. Company profits

### 28.2 4 Types of Crowdsourcing (Brabham, 2011)

#### 1. Knowledge Discovery & Management:

- Organization tasks crowd with finding/collecting information
- Examples: Peer-to-Patent, SeeClickFix

#### 2. Broadcast Search:

- Organization tasks crowd with solving empirical problems
- Examples: InnoCentive, Goldcorp Challenge

#### 3. Peer-Vetted Creative Production:

- Organization tasks crowd with creating/selecting creative ideas
- Examples: Threadless, Doritos Crash the Super Bowl

#### 4. Distributed Human Intelligence Tasking:

- Organization tasks crowd with analyzing large amounts of information
- Examples: Amazon Mechanical Turk

### 28.3 Why People Engage in Crowdsourcing

- Earn money
- Develop creative skills
- Network with creative professionals
- Build portfolio for future employment
- Challenge oneself
- Socialize and make friends
- Pass time when bored
- Contribute to large project of common interest

- Share with others
- Have fun

### 28.4 Free and Open Source Software (FOSS)

#### Richard Stallman & GNU (1980s):

- Refused access to Xerox laser printer source code (1980)
- Convinced people should be free to modify all software
- Founded Free Software Foundation (FSF) in 1985
- Created GNU project: "GNU's Not Unix"
- Philosophy: "technical means to a social end"

#### Free Software - 4 Freedoms:

0. Freedom to run the program for any purpose
1. Freedom to study how program works and change it
2. Freedom to redistribute copies to help others
3. Freedom to distribute copies of modified versions

#### Open Source (Eric Raymond, 1998):

- "Free software" was ambiguous (confused with freeware)
- Focused on pragmatic benefits vs ideological
- Key Principle: "Given enough eyeballs, all bugs are shallow"

### 28.5 Free Software vs Open Source

Free Software	Open Source
Social movement	Development methodology
User rights/freedoms	Pragmatic benefits
Narrower licenses	Wider licenses
Less proprietary mixing	More proprietary mixing

### 28.6 FOSS License Spectrum

#### From Permissive to Restrictive:

Public Domain → MIT → BSD → Apache → GPLv2 → GPLv3 → AGPL

#### Public Domain:

- No intellectual property rights
- Rarely used for software

#### MIT License:

- User can do anything with software
- Must maintain original copyright notice
- Very permissive

#### BSD Licenses (2, 3, 4-clause):

- Similar to MIT
- Must acknowledge use if redistributing
- Very permissive

#### Apache Software License:

- Similar to BSD but addresses patent rights
- Permissive

#### GNU GPL (General Public License):

- **Copyleft** license - more restrictive
- Can use and modify code

- MUST release all modified code under same license
- Any code that "touches" it must also be GPL
- Ensures derivative works remain free

#### AGPL (Afferro GPL):

- Even more restrictive than GPL
- Designed for network/web services
- Must share source code even for software used over network

### 28.7 Copyleft

- Play on "copyright"
- Method for making program free AND requiring all modified/extended versions to be free
- Example: GNU Public License (GPL)
- Symbol: (reversed C in circle)

### 28.8 Creative Commons

Standardized way to grant copyright permissions for creative/academic works

License Types (from least to most restrictive):

- CC BY (Attribution)
- CC BY-SA (Attribution-ShareAlike)
- CC BY-NC (Attribution-NonCommercial)
- CC BY-ND (Attribution-NoDerivatives)
- CC BY-NC-SA (Attribution-NonCommercial-ShareAlike)
- CC BY-NC-ND (Attribution-NonCommercial-NoDerivatives)

### 28.9 Open Source Business Models

1. Sell Support and Services (e.g., Canonical/Ubuntu)
2. Sell Certified Version (e.g., Cloudera/Hadoop)
3. Sell "Enterprise Edition" (e.g., MySQL)
4. Dual Licensing (Copyleft for free, proprietary for modifications)
5. Other Advantages (e.g., Google/Android ecosystem)
6. Freemium Model (e.g., Elementary OS "pay what you want")

### 28.10 OpenStreetMap Case Study

- Initiative to create/provide free geographic data
- "Wikipedia of maps"
- Crowd sourcing + Open source + User innovation
- 2M+ users collecting data
- Methods: manual survey, GPS, aerial imagery
- Open data license
- Innovations: KartaView, OSRM, Pokémon Go uses OSM

## 29 26C. Week 6 - Platform Details

### 29.1 Lead Users (von Hippel, 1986)

#### Characteristics:

1. Face needs months/years BEFORE general market realizes

them

- 2. Benefit significantly from obtaining a solution
- 3. Spend resources trying to solve those needs
- 4. At leading edge of trends, very knowledgeable about "state of the art"

**Important:** Lead users are NOT usually a company's "lead customers". They're typically dissatisfied with current products and have created their own solutions.

## 29.2 User Innovation Examples

### World Wide Web:

- Created by researcher for scientists to communicate

### Firefox Add-ons:

- Developers create plugins for own use, then share

### Apache Server Modules:

- Originally by web server administrator

### MySQL Case Study:

- Started wanting to use mSQL database

- mSQL not fast/flexible enough

- Created new SQL interface with similar API to mSQL

- Named after co-founder's daughter (My)

- Sun Microsystems bought for \$1B (2008)

### Slack Case Study:

1. Game Neverending (failed) → Built Flickr using the technology

2. Yahoo acquired Flickr (2005)

3. Glitch game (failed again)

4. Built team communication tool for coordinating work

5. Realized tool could help other teams

6. Slack launched → IPO 2019, valuation >\$20B

## 29.3 Maker Movement

- "Umbrella term for independent inventors, designers and tinkerers"

- Make Magazine (first published 2005)

- Maker Faire: First SF Bay Area 2006, 200 faires worldwide by 2016

- IT innovation → simple to start

- Physical building → becoming easier (3D printers, Raspberry Pi)

## 29.4 Alternative Perspective on User Innovation

### Criticisms:

1. Can't predict future demand - users don't know if they'll like breakthrough products

2. Misses disruptive innovations - focus on users leads to incremental

3. Leads to sameness - user insights widely available

**Apple's Approach (Steve Jobs):** "We figure out what we

want. We're pretty good at having the right discipline to think through whether a lot of other people are going to want it too."

## 29.5 Platform Ecosystem - 4 Main Roles

1. **PRODUCERS:** Creators of platform's offerings (e.g., App developers on Android)
2. **CONSUMERS:** Buyers or users of offerings (e.g., App users)
3. **PROVIDERS:** Interfaces for the platform (e.g., Mobile devices on Android)
4. **OWNER:** Controls platform IP, arbiters of who participates (e.g., Google owns Android)

Note: Players may shift rapidly from one role to another

## 29.6 Pipelines vs Platforms

Pipeline Business	Platform Business
Linear flow	Multi-directional
Take in resources	Create value by controlling interactions
Add value	Between producers and consumers
Release higher-value products	Orchestrate interactions
Centrally employed staff	External parties
Owned assets	

## 29.7 Platform Strategy Shifts

### From Pipelines to Platforms:

1. Resource Control → Resource Orchestration (main asset: network)
2. Internal Optimization → External Interaction (focus: facilitating network)
3. Customer Value → Ecosystem Value (focus: total value of expanding network)

## 29.8 Measuring Platform Businesses

### 4 Key Metrics:

1. **Interaction Failure:** Failed interactions between producers/consumers
2. **Engagement:** Level of participation enhancing network effects
3. **Match Quality:** Quality of producer-consumer interactions
4. **Negative Network Effects:** Over-supply or over-demand issues

## 29.9 Types of Platform Businesses

### 1. Integrator Platform:

- Example: Apple iPhone app architecture
- External innovators → Platform → Customers
- High control by platform

### 2. Product Platform:

- Example: Intel x86 Architecture
- Platform → External Innovators → Customers
- High control by platform

### 3. Two-Sided Platform:

- Example: Airbnb (property owners travelers)
- Platform connects External Innovators Customers
- High autonomy of external parties

## 29.10 Common Platform Types (Deloitte, 2015)

### 1. Aggregation Platforms:

- Facilitate transactions
- Connect users to resources
- Hub-and-spoke model
- Examples: eBay, Etsy, App Store, Airbnb, Kaggle

### 2. Social Platforms:

- Facilitate social interactions
- Connect individuals to communities
- Mesh relationship networks
- Examples: Facebook, Twitter

### 3. Mobilization Platforms:

- Facilitate mobilization
- Move people to act together
- Foster long-term relationships for shared goals
- Examples: Linux, Apache

## 29.11 Platform Value Areas

1. **Search:** Help users find products/services/participants (filters, algorithms, recommendations)
2. **Trust:** Reputation systems, ratings/reviews, verification, secure messaging
3. **Financial Transactions:** Secure payment processing, escrow, transaction management
4. **Physical/Digital Delivery:** Logistics, order tracking, downloads/streaming

## 29.12 Platform Monetization Strategies

### Direct Revenue:

1. Sales Commission (% of each transaction)
2. Subscription Fees (regular access fees, e.g., Prime)
3. Transaction-Based (per-use charges)
4. Listing Fees (charges to post offerings)
5. Fulfillment Fees (logistics services)

### Indirect Revenue:

1. Advertising (promoted listings, display ads)
2. Data Monetization (analytics, insights, trends)
3. Sponsorships (featured placements)
4. Recommendation Algorithms (using user data)

## 29.13 Case Study: Apple iPhone Platform

### The Disruption (2007-2015):

- 2007: Nokia, Samsung, Motorola, Sony Ericsson, LG = 90% global profits
- 2015: iPhone alone = 92% global profits
- Former incumbents made no profit (except Samsung)

### Platform Strategy:

- iPhone + iOS as more than product/service
- Designed to connect two-sided market: App developers / App users
- Value increases as participants grow (network effects)

### Results:

- January 2025: 1.92 million apps available
- 2022: \$1.1 trillion in total billings/sales in App Store ecosystem

## 29.14 Case Study: League of Legends & Fortnite

### League of Legends (Riot Games, 2009):

- Strategy: Build a platform, not just a game
- 67 million players/month
- \$1B annual revenue (2009), \$1.75B (2020)
- Free to play, monetize via: character skins, boosters, live events, e-sports, broadcasting, sponsorships

### Fortnite (Epic Games):

- Similar platform model
- Creator Economy 2.0
- Unreal Editor for Fortnite - user-generated content platform

## 30 26D. Week 7 - Lean Startup Details

### 30.1 Unicorn Companies

#### Definitions:

- **Unicorn:** Privately held startup valued at \$1+ billion
- **Decacorn:** Privately held startup valued at \$10+ billion
- **Undercorn:** Unicorns that sell/IPO below last private valuation

#### Key Statistics (as of 2024):

- Over 1,200 global unicorns worth \$3.86 trillion
- Top countries: USA (561), China (173), India (65)
- Top industries: Fintech, Internet Software, E-commerce, AI

#### Five Primary Business Models:

1. E-Commerce (36%): Consumer pays for goods/services online (Uber, Airbnb)
2. Audience (27%): Free to use, makes money through ads/leads (Snapchat)
3. Enterprise Software (20%): Business pays for larger-scale software

4. SaaS (12%): Cloud-based software via freemium/monthly model (Slack, MongoDB)
5. Consumer Electronics/IoT (6%): Physical products (Xiaomi)

### 30.2 Startups vs Traditional Companies

Aspect	Established	Startup
Markets	Known	Mostly unknown
Customers	Known	Mostly unknown
Products	Known	Mostly unknown
Future Features	Learn from customers	Test hypotheses
Business Model	Execute current	Search for best
Product	Full specifications	MVP
Development	Smooth execution	Pivots
Structure	Relatively stable	Fluid

### 30.3 Traditional Business Plan - 9 Deadly Sins

1. "I know what the customer wants" assumption
2. "I know what features to build" flaw
3. Focus on launch date instead of learning
4. Emphasis on execution instead of testing hypotheses
5. No room for trial and error
6. Wrong job titles - traditional roles don't fit startup needs
7. Sales/marketing execute a plan rather than discover customers
8. Presumption of success leads to premature scaling
9. Management by crisis leads to death spiral

### 30.4 14 Rules of Customer Development Manifesto

1. **Get Outside the Building:** No facts inside your building
2. Pair with Agile Development
3. Failure is integral - part of the search process
4. Make continuous iterations and pivots
5. No business plan survives first contact - use Business Model Canvas
6. Design experiments to test hypotheses
7. Market type changes everything (5 types)
8. Different metrics for startups (not revenue/profit initially)
9. Fast decision-making - speed, cycle time, tempo matter
10. Passion is critical - founders must be passionate
11. Different job titles - startup roles ≠ corporate roles
12. Preserve cash - spend only when validated
13. Communicate and share learning - entire team must learn
14. Get buy-in - success requires team commitment

### 30.5 Agile Manifesto - 4 Values & 12 Principles

#### 4 Core Values:

1. Individuals and interactions > Processes and tools

2. Working software > Comprehensive documentation
3. Customer collaboration > Contract negotiation
4. Responding to change > Following a plan

#### 12 Principles:

1. Satisfy customer through early and continuous delivery
2. Welcome changing requirements, even late in development
3. Deliver working software frequently (weeks to months)
4. Business people and developers work together daily
5. Build projects around motivated individuals
6. Face-to-face conversation is most efficient
7. Working software is primary measure of progress
8. Sustainable development - constant pace indefinitely
9. Technical excellence and good design enhance agility
10. Simplicity - maximize work not done
11. Self-organizing teams produce best results
12. Regular reflection and adjustment for effectiveness

### 30.6 3 Agile Methodologies Comparison

Methodology	Principles	Best For
SCRUM	Empower creative cross-functional teams	Creative cultures with high trust, Radical innovation
KANBAN	Visualize workflows, limit work in process	Process-oriented cultures, evolutionary improvements
LEAN DEV	Eliminate waste from system as whole	Process-oriented, evolutionary improvements

### 30.7 5 Principles of Lean Startup

1. **Entrepreneurs are everywhere:** Don't need garage to be in startup
2. **Entrepreneurship is management:** Startup is institution requiring new management
3. **Validated learning:** Learn how to build sustainable business through experiments
4. **Innovation accounting:** Measure progress, set milestones, prioritize work
5. **Build-Measure-Learn:** Turn ideas → products, measure customer response, learn whether to pivot or persevere. All processes should accelerate this loop.

### 30.8 Build-Measure-Learn Loop Detailed

#### Learn Faster:

- Split tests, customer interviews
- Customer development, Five Whys root cause analysis
- Customer advisory board, falsifiable hypothesis
- Product owner accountability, cohort analysis

- Cross-functional teams, smoke tests

#### Build/Code Faster:

- Unit tests, usability tests
- Continuous integration, incremental deployment
- Free & open-source components, cloud computing
- Cluster immune system, just-in-time scalability
- Refactoring, developer sandbox

#### Measure Faster:

- Split tests, clear product owner
- Continuous deployment, usability tests
- Real-time monitoring, customer liaison
- Funnel analysis, cohort analysis
- Net promoter score, search engine marketing
- Real-time alerting, predictive monitoring

### 30.9 Product-Market Fit Pyramid (Dan Olsen)

#### The Pyramid (Bottom to Top):

1. *Market Layer - Target Customer*: Determine who exactly you're building for
2. *Market Layer - Underserved Needs*: Identify underserved customer needs
3. *Product Layer - Value Proposition*: Define how you'll solve problems
4. *Product Layer - MVP Feature Set*: Specify minimum features delivering value
5. *Product Layer - MVP Prototype*: Create MVP prototype
6. *Product Layer - Test with Customers*: Test MVP with customers

## 31 26E. Week 8-9 - BMC & Funding Details

### 31.1 Customer Development - Detailed 4 Phases

#### Phase 1: Customer Discovery

- State problem and solution hypothesis
- Test the problem - Get out of building, build wireframe/prototype
- Test the solution - Build low-fidelity MVP, test with customers
- Verify or Pivot - Check if problem high-value, business model ready, market big enough

#### Phase 2: Customer Validation

- Get ready and sell - Acquire/activate customers, build high-fidelity MVP
- Test sell - Sell to evangelists and early adopters
- Develop positioning - Based on customer feedback
- Verify repeatability - Ready to scale marketing and sales?

**Product/Market Fit** = Match between Value Proposition and Customer Segment

#### Phase 3: Customer Creation

- Build end-user demand
- Build sales channel
- Scale business

#### Phase 4: Company Building

- Transition from startup to typical company

### 31.2 Agile Success Rates (Standish CHAOS Report)

Method	Successful	Challenged	Failed
Agile	39%	52%	9%
Waterfall	11%	60%	29%

### 31.3 Business Model Canvas - 9 Blocks Detailed

#### Block 1: Customer Segments

5 Types:

1. *Mass Market*: Broad customer base (e.g., Facebook)
2. *Niche Market*: Specific specialized segments (e.g., Qualcomm)
3. *Segmented*: Slightly different needs (e.g., Apple - MacBook Air, Pro, iMac)
4. *Diversified*: Unrelated segments (e.g., Amazon retail + AWS)
5. *Multi-sided Platforms*: Two+ interdependent segments (e.g., Airbnb)

#### Block 2: Value Propositions

- Pain Relievers: How products alleviate customer pains
- Gain Creators: How products create customer gains
- Value Pyramid: Functional → Emotional → Life Changing → Social Impact

#### Block 3: Channels

5 Channel Phases:

1. Awareness: How raise awareness?
2. Evaluation: How help customers evaluate?
3. Purchase: How allow purchase?
4. Delivery: How deliver value proposition?
5. After Sales: How provide post-purchase support?

#### Block 4: Customer Relationships

Types:

- Personal Assistance (e.g., Salesforce)
- Dedicated Personal Assistance (e.g., private banking)
- Self-Service (e.g., Slack)
- Automated Services (e.g., Netflix recommendations)
- Communities (e.g., Envato)
- Co-Creation (e.g., YouTube, Amazon reviews)

#### Block 5: Revenue Streams

Types:

1. Asset Sale: Selling ownership (e.g., Tesla cars)
2. Usage Fee: Pay per use (e.g., IBM Watson)
3. Subscription Fee: Continuous access (e.g., Spotify, Asana)

- 4. Lending/Renting/Leasing: Temporary use (e.g., Zipcar)

- 5. Licensing: Permission to use IP

- 6. Transaction/Brokerage Fees: Intermediation (e.g., PayPal, Stripe)

- 7. Advertising: Fees for advertising (e.g., Google Ads)

#### Block 6: Key Activities

- Most important actions to make business model work
- Examples: Software development (Microsoft), Supply chain (Dell), Platform development

#### Block 7: Key Resources

Types:

1. Technology: Proprietary/open-source software
2. Data: Training data, user data (crucial for AI)
3. Human: Knowledge workers, scientists
4. Intellectual: Patents, copyrights
5. Physical: Manufacturing facilities

Financial: Cash, debt, stock options

#### Block 8: Key Partnerships

4 Types:

1. Strategic alliances between non-competitors
2. Cooperation between competitors
3. Joint ventures to develop new businesses
4. Buyer-supplier relationships for reliable supplies

Reasons: Optimize business models, reduce risk, acquire resources

#### Block 9: Cost Structure

Types:

- Fixed Costs: Remain same regardless of volume (salaries, rent)
- Variable Costs: Vary with volume

### 31.4 Funding Rounds Progression

Typical Progression:

1. **Pre-Seed/Seed**: Friends & family ~\$15K, Seed Round \$200K at \$1M valuation
2. **Series A**: \$2M at \$4M valuation - validate product-market fit
3. **Series B, C, D....**: Increasing amounts - scale operations, expand markets
4. **IPO**: \$235M at \$2.6B valuation - become publicly traded

#### Example - Canva:

- Seed (2013): \$3.6M
- Series A (2015): \$15M
- Series B (2016): \$15M
- Series C (2018): \$40M
- Series D (2019): \$70M
- Series E (2019): \$85M
- Total: \$572.6M over 14 rounds

### 31.5 Capital Raising Process (5 Steps)

1. **Initial Meeting:** Present pitch, follow up
2. **Basic Analysis:** Review revenue plans, understand 3-year plan, meet portfolio companies
3. **Due Diligence:** Cohort analysis, customer calls, reference calls, product roadmap, technical review
4. **Partner Meeting:** Present to investment committee (if pass)
5. **Term Sheet:** Negotiate terms, financial/legal diligence, key employee references

### 31.6 Pitch Format (Melbourne Accelerator Program)

8-Section Structure:

1. **Introduction:** Introduce yourself, one-sentence elevator pitch, traction teaser
2. **Problem/Opportunity:** What problem solving? Deep understanding?
3. **Your solution:** Is solution well validated?
4. **Market:** Who uses product? Why need it? How big? Why now?
5. **Traction:** Progress to date, key metrics, validation
6. **Competition:** What else out there? Why better? Competitive advantages?
7. **Business Model:** Revenue/funding model, how grow and scale?
8. **Team:** Why will founders succeed? How well work together?

### 31.7 Alternative Funding - MVP Ventures Program (NSW)

#### Grant Details:

- Amount: \$25,000 to \$200,000
- For: Commercialization of products/services
- Must be progressing TRL 3-9
- Requires 50% matched funding
- Must complete within 12 months

#### Eligibility:

- Related to commercialization
- Activities in NSW
- Aligned with NSW Industry Development Framework
- Demonstrate why other funding can't be accessed
- Demonstrate matched 50% funding

## 32 27. Startup Failures & Valley of Death

### 32.1 Top 3 Reasons Startups Fail

#### 1. No Market Need (42%)

- Building product/service with no significant demand
- Why: Founders caught up in solution, don't validate problem
- Assume innovation = success without testing
- Build before customer demand validated
- **Prevention:** Validate pain points FIRST, test willingness to pay, build MVP

#### 2. Running Out of Cash (29%)

- Underestimate costs or overestimate revenue growth
- **Valley of Death:** Period between launch and revenue generation
- High early costs (R&D, team, infrastructure) before PMF
- Causes: Underestimate burn rate, overestimate sales cycles, fail to raise sufficient capital
- Deep-tech especially challenging (hardware, long R&D)

#### 3. Team/Founding Issues (23%)

- Team dysfunction, lack of key skills, founder conflicts
- Specific: Lack of trust, missing diverse expertise, poor decision-making, equity disputes
- Consequences: Slow execution, strategic drift, missed opportunities
- Paul Graham: Team quality often determines success/failure

### 32.2 The J-Curve

- Massive losses in early-stage startups
- Revenue << Expenses = Loss
- Key points: Burn rate, max financing needs, first cash flow positive, breakeven
- Most startups die in valley before reaching profitability
- Examples: Amazon (1994-2001), Uber (\$5.2B quarterly loss), Paytm (\$549M loss on \$448M revenue)

## 33 28. Organizational Culture & Innovation

### 33.1 Taylorism (Scientific Management - 1911)

**Creator:** Frederick Winslow Taylor

**Core Principles:**

1. Standardize work into separate, distinct tasks
2. Scientifically study each task for optimization
3. Continually measure performance of each task
4. Compensate workers based on performance metrics

**Organizational Influence:**

- Mechanistic hierarchies with formal lines
- Top-down decision making

- Detailed role definitions & procedures
- Emphasis on efficiency, control, standardization

#### Impact on Innovation:

- *Good for:* Existing operations, cost reduction, process improvement, sustaining innovation
- *Bad for:* Radical innovation, experimentation, creativity, breakthrough ideas
- Reason: Innovation needs flexibility, experimentation, cross-functional collaboration - opposite of rigid control

### 33.2 Steven Johnson - Where Good Ideas Come From

#### 1. Liquid Networks

- Ideas need to collide and connect
- Dense networks of diverse people sharing ideas
- Physical spaces that enable serendipitous encounters
- Open collaboration > isolated genius
- Example: Enlightenment coffeehouses, university campuses, open office spaces

#### 2. The Slow Hunch

- Most innovations aren't "eureka moments"
- Ideas form slowly over time, require nurturing
- Need to document hunches, revisit them
- Connecting slow hunches from different people creates breakthroughs

#### 3. Connecting Ideas

- Innovation = connecting existing ideas in new ways
- Cross-disciplinary thinking essential
- Diverse perspectives spark innovation
- "Adjacent possible" - innovations build on what exists

### 33.3 Innovation Spaces Matrix (Johnson)

	Individual	Networked
Market	1st Quadrant (weak innovation)	4th Quadrant (moderate)
Non-Market	2nd Quadrant (moderate)	3rd Quadrant <b>MOST innovation</b>

**Key Insight:** Most breakthrough innovations come from **networked, non-market** spaces (universities, open-source communities, research labs)

## 34 29. Organizational Structure

### 34.1 Mechanistic vs Organic Structures

**Mechanistic** (Machine-like):

- *High formalization:* Documented procedures, standardized processes
- *High standardization:* Uniform methods across organization

- *High centralization:* Top-down decision making
- *Narrow spans of control:* Many management layers
- **Best for:** Stable environments, routine tasks, efficiency focus

**Organic** (Living organism):

- *Low formalization:* Flexible, few written rules
- *Low standardization:* Customized approaches
- *Low centralization:* Decentralized decision-making
- *Wide spans of control:* Flat hierarchy
- **Best for:** Dynamic environments, innovation, creativity
- **Examples:** Startups, R&D labs, creative agencies

### 34.2 Key Dimensions

**Formalization:**

- Extent to which rules, procedures documented
- High = bureaucratic, low = flexible

**Standardization:**

- Extent to which activities performed uniformly
- High = consistent, low = customized

**Centralization:**

- Where decision-making authority resides
- High = top management decides, low = delegated

### 34.3 Implications for Innovation

**Mechanistic Structures:**

- Enable sustaining innovation (incremental improvements)
- Struggle with radical innovation (too rigid)
- Good for process innovation in manufacturing

**Organic Structures:**

- Enable radical innovation (flexibility, experimentation)
- May struggle with efficiency at scale
- Good for product innovation, new markets

## 35 30. Ambidextrous Organizations

### 35.1 Definition

Organizations that simultaneously **explore** (new opportunities) AND **exploit** (existing capabilities)

### 35.2 The Challenge

- *Exploration:* Needs experimentation, risk-taking, flexibility, long-term view (organic structure)
- *Exploitation:* Needs efficiency, refinement, execution, short-term focus (mechanistic structure)
- These require contradictory structures & cultures
- Most companies struggle to do both

### 35.3 Solutions

**Structural Separation:**

- Create separate units for exploration vs exploitation

- Different structures, metrics, incentives for each
- Example: Google X (moonshots) vs Google Search (optimization)

#### **Contextual Ambidexterity:**

- Employees switch between exploration & exploitation based on context
- Requires supportive culture & systems
- Example: 20% time policies, hackathons

## **36 31. Company Size & Innovation**

### **36.1 Large Company Advantages**

- Greater resources (R&D budgets, facilities, talent)
- Established market position & brand
- Economies of scale
- Risk tolerance for big bets
- Complementary assets (distribution, manufacturing)

### **36.2 Large Company Disadvantages**

- **Bureaucratic inertia:** Resistance to change, slow decision-making
- Established structures favor incremental over radical innovation
- Multiple approval layers slow experimentation
- Risk-averse culture (protecting existing business)
- Internal politics & turf battles
- "Not invented here" syndrome
- Difficulty pivoting quickly

### **36.3 Small Company/Startup Advantages**

- Agility: Fast decision-making, quick pivots
- Flexibility: Can experiment without bureaucracy
- Hunger: Desperate to succeed, willing to take risks
- Focus: Single product/market, undivided attention
- Culture: Entrepreneurial, innovation-focused
- Less to lose: Can disrupt without cannibalizing

### **36.4 Small Company Disadvantages**

- Limited resources (cash, people, time)
- Lack of established brand/credibility
- Difficulty crossing chasm to mainstream
- Missing complementary assets
- Higher failure risk

## **37 32. Innovation in Practice: Case Studies**

### **37.1 Skunk Works (Lockheed Martin)**

**Creator:** Kelly Johnson (1943)

**Concept:** Small, autonomous team for breakthrough projects, separate from bureaucracy

**Kelly's 14 Rules (Key Selections):**

1. **Rule 1:** Manager has practically complete control, reports to division president or higher
2. **Rule 3:** Restrict personnel viciously - use 10-25% compared to "normal" systems
3. **Rule 4:** Simple drawing system with great flexibility for changes
4. **Rule 11:** Timely funding so contractor doesn't run to bank
5. **Rule 14:** Reward performance NOT based on # supervised

#### **Results:**

- U-2 spy plane, SR-71 Blackbird, F-117 stealth fighter
- Faster, cheaper, more innovative than traditional approach
- Model copied by many companies

### **37.2 Microsoft - Culture Change (Satya Nadella)**

**Problem:** Steve Ballmer era - competitive, zero-sum culture, silos, "I know it all" attitude

#### **Nadella's Changes:**

- **Growth Mindset:** From "know-it-all" to "learn-it-all"
- Culture as explicit conversation: Talk about it constantly
- Hackathons: Break down silos, enable collaboration
- Fewer, bigger bets: Focus resources
- Partner-positive: Collaborate instead of compete
- Customer obsession: Outside-in thinking

#### **Results:**

- Market cap: \$300B (2014) → \$3T (2024)
- Azure cloud success, GitHub acquisition, OpenAI partnership
- Cultural transformation enabled technical innovation

### **37.3 Apple - Macintosh Renegade Team (1980s)**

#### **Context:**

- Apple producing Apple II for technical users
- Corporate environment structured for manufacturing

#### **Jobs' Vision:**

- Intuitive computer for everyone (not just technical)
- Based on mouse + desktop icons from Xerox PARC

#### **The Solution - Renegade Culture:**

- New product development group for Macintosh
- Consciously different from mainstream practices
- Hidden in building OFF main campus
- "Hijacked" best people from another project
- Different culture with more flexibility

**Lesson:** Radical innovation may require separate culture/structure

## **37.4 Google - Maintaining Small Company Benefits**

**Eric Schmidt's Philosophy:** "Avoid divisional structure that prevents collaboration. If people understand company values, they should self-organize."

#### **1. Google's "20% Time":**

- Encourages employees to spend 20% time on what benefits Google
- Success Stories: Gmail, Google News, AdSense
- Reality: ~10% actually use it (Laszlo Bock, 2015)
- Why It Matters: "Idea of 20% time more important than reality"

#### **2. X (Moonshot Factory):**

- Run at hardest parts of problem, try to prove can't be done
- Actively embrace failure: mistakes = progress
- **3 Requirements:** Big Problem + Radical Solution + Breakthrough Tech
- Tackle with research riskiness + startup speed
- Examples: Waymo (self-driving), Loon (internet balloons), Wing (delivery drones)

## **38 33. Judging IT Innovations**

### **38.1 Why Judging Matters**

- Many innovations released daily without evidence of success
- Need framework to evaluate potential value
- Understand own innovations
- Make investment decisions
- Compare competing solutions objectively

### **38.2 5 Standard Judging Criteria**

#### **1. TECHNOLOGY**

- Engineering qualities
- Availability, security, safety
- Reliability, performance, scalability
- Technical implementation quality

#### **2. DESIGN**

- Aesthetic & user experience
- Visual, auditory, perceptual design
- Intuitive interface
- Usability, accessibility

#### **3. INNOVATION**

- Novelty & uniqueness
- New product/service or approach
- Novel features or combinations
- Breakthrough vs incremental

#### **4. FEASIBILITY**

- Business viability
- Business model sustainability

- Market competitiveness
- Resource requirements realistic

## 5. CONCEPT/IMPACT

- Value & reach
- User value proposition
- Broad appeal potential
- Global impact, social benefits

### 38.3 Major Innovation Competitions

#### Microsoft Imagine Cup:

- World's premier student tech competition
- 20+ years, dozens of countries
- \$100,000 prize for World Champion
- Criteria: 5 standard criteria
- Encourages MS technology use
- Think "blue sky" - big claims

#### Harvard i3:

- Harvard's premier startup competition
- Year-long program with grants & incubator space
- **3 Core Criteria:** IMPACT (innovative, measurable), PLAN (realistic, financially sustainable), TEAM (skills, commitment, complementary)

#### CES Innovation Awards:

- Consumer Electronics Show
- World's gathering for consumer tech
- Judging Panel: Independent designer, engineer, trade press
- Criteria: 5 standard criteria
- Special: Accessible Technologies category

### 38.4 Judging Best Practices

#### Panel Composition:

- Diverse pool: Business executives, industry experts, academics
- Complementary expertise
- Independence from entrants

#### Critical Considerations:

- Judge training CRITICAL for uniformity

- Judgment subjective but criteria enable fair comparison
- Criteria varies by category/event
- Calibration sessions important

#### Judge Incentives:

- Giving back (most committed)
- Professional networking
- Professional courtesy (least reliable)

## 39 34. Week 10/11 Exam Focus

### 39.1 Critical Concepts to Know

#### Startup Failures:

- Top 3: No market need, cash starvation, team issues
- Valley of Death = critical period between launch & revenue
- Must validate before building
- Team quality = success predictor

#### Culture for Innovation:

- Taylorism ≠ Innovation (too rigid)
- Need: Liquid networks, slow hunch, connecting ideas
- Networked + Non-market = most innovation
- Appetite for risk essential
- Bold bets + rapid pivots

#### Structure for Innovation:

- Organic for R&D, mechanistic for operations
- Ambidextrous = explore + exploit simultaneously
- Small teams, empowered decisions
- Skunk works for breakthroughs
- Avoid divisional silos

#### Large Company Challenges:

- Bureaucratic inertia & slow decisions
- Established structures favor incremental
- Risk-averse culture protecting existing
- Internal politics slow innovation

#### Judging Framework:

- 5 criteria: Technology, Design, Innovation, Feasibility, Concept/Impact

- Different competitions emphasize different aspects
- Panel diversity & training critical
- Objective evaluation despite subjectivity

### 39.2 Key Case Studies

#### Apple Vision Pro (Exam Scenario):

- Mixed-reality headset, spatial computing
- Potential disruptor in computing/wearables
- Analyze using large company disadvantages
- Consider bureaucratic inertia, risk aversion, internal challenges

#### DAQRI (AR Startup Failure):

- Raised hundreds of millions, shut down 2019
- Mistakes: No market need validation, extreme cash burn
- Lessons: Deep-tech needs longer runway, hardware + enterprise = high risk

### 39.3 Exam Answer Approach

#### For "Large Company Disadvantages" Questions:

1. Name specific disadvantage clearly
2. Explain the mechanism/why it happens
3. Show impact on innovation-driven markets
4. Apply to scenario (e.g., Vision Pro)
5. Use keywords: bureaucratic inertia, established structures, risk-averse

#### For "Culture & Innovation" Questions:

1. Reference Taylorism (mechanistic, bad for innovation)
2. Contrast with organic, networked approaches
3. Cite Johnson's concepts (liquid networks, slow hunch)
4. Give examples: Skunk Works, Google 20%, Microsoft culture change

#### For "Judging Innovation" Questions:

1. State all 5 criteria
2. Explain relevance of each to scenario
3. Show how criteria enable objective comparison
4. Reference specific competition if relevant