TECHNOLOGY ASSESSMENT

EXAMPLE COMPANY

TABLE OF CONTENTS

1. Project Management
2. Product Management
3. Alignment and Engagement
4. Software Development
5. Systems and Infrastructure
6. Tools
7. Security
8. Org Structure
9. People
10. The CTO Role

PART 1

PROJECT MANAGEMENT

PROJECT MANAGEMENT

- Description of current project management processes

STRENGTHS OF CURRENT SYSTEM

- Example: Low overhead
- Example: Few barriers between developers and business owners

WEAKNESSES OF CURRENT SYSTEM

- Example: Lack of communication across teams and sometimes within teams
- Example: Lack of ownership

OPPORTUNITIES OF CURRENT SYSTEM

- Example: We can manage time much better and immediately be far more efficient

THREATS OF CURRENT SYSTEM

- Example: Lack of standard development processes across groups could impact coordination and quality controls as team scales

INTRODUCTION TO DIFFERENT APPROACHES

- If valuable, an overview of various project management methodologies.

AGILE OVERVIEW

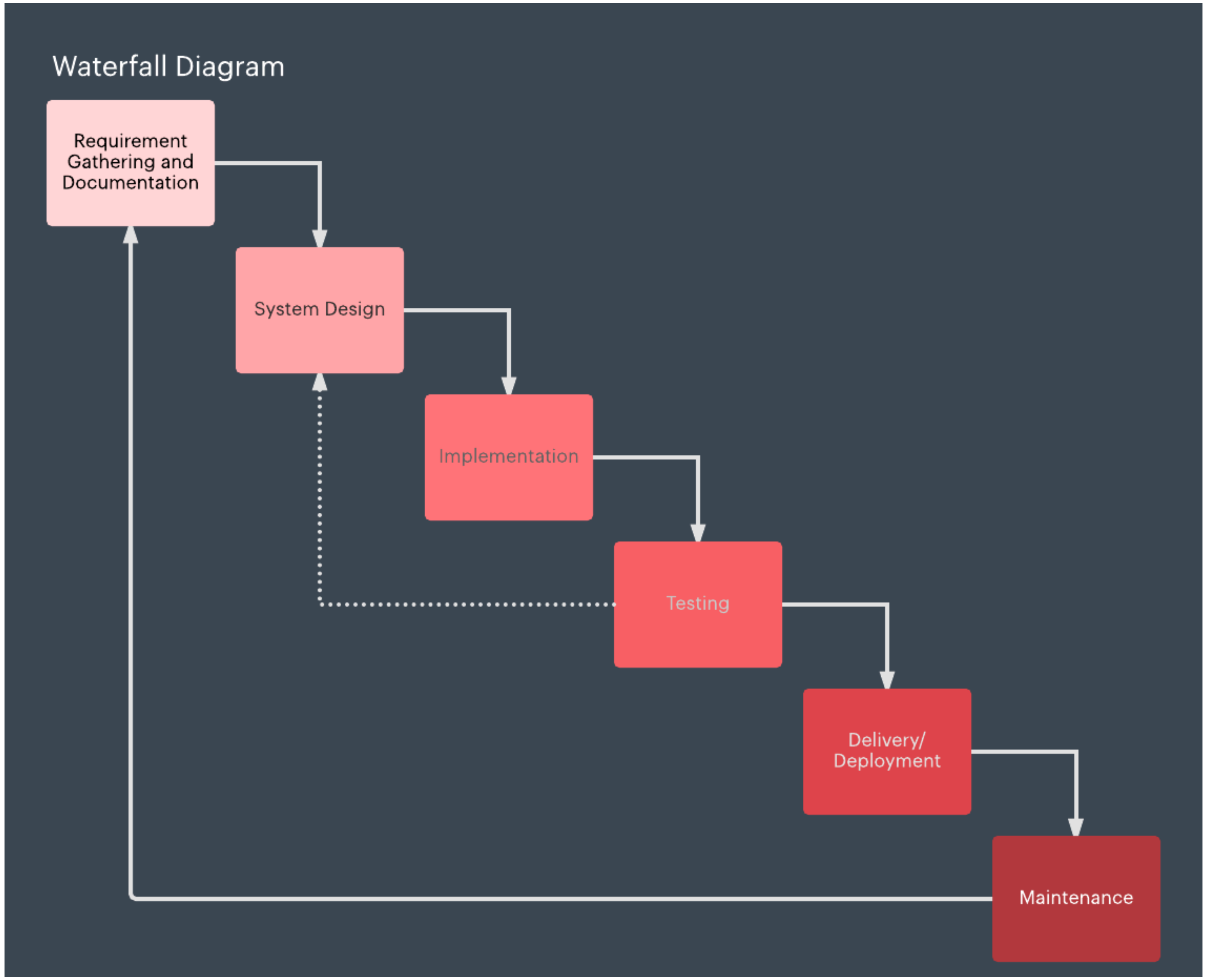
Agile Manifesto

We are uncovering better ways of developing software by doing it and helping others do it. Through this work we have come to value:

- Individuals and interactions over processes and tools
- Working software over comprehensive documentation
- Customer collaboration over contract negotiation
- Responding to change over following a plan

That is, while there is value in the items on the right, we value the items on the left more.

WATERFALL


Waterfall Diagram
Requirement Gathering and Documentation
System Design
Implementation
Testing
Delivery/ Deployment
Maintenance

AGILE

- Scrum and Kanban are specific implementations of agile
- Agile is a broad approach to flexible development, focusing on product quality
- Adaptive, simultaneous workflows
- Incremental work
- Similar to waterfall but with short iterations, continuous improvement feedback loop, focus on shipping

TWELVE PRINCIPLES OF AGILE DEVELOPMENT

1. Our highest priority is to satisfy the customer through early and continuous delivery of valuable software.
2. Welcome changing requirements, even late in development. Agile processes harness change for the customer's competitive advantage.
3. Deliver working software frequently, from a couple of weeks to a couple of months, with a preference to the shorter timescale.
4. Business people and developers must work together daily throughout the project.
5. Build projects around motivated individuals. Give them the environment and support they need, and trust them to get the job done.
6. The most efficient and effective method of conveying information to and within a development team is face-to-face conversation.
7. Working software is the primary measure of progress.
8. Agile processes promote sustainable development. The sponsors, developers, and users should be able to maintain a constant pace indefinitely.
9. Continuous attention to technical excellence and good design enhances agility.
10. Simplicity--the art of maximizing the amount of work not done--is essential.
11. The best architectures, requirements, and designs emerge from self-organizing teams.
12. At regular intervals, the team reflects on how to become more effective, then tunes and adjusts its behavior accordingly.

TWELVE PRINCIPLES OF AGILE DEVELOPMENT AT EXAMPLE COMPANY

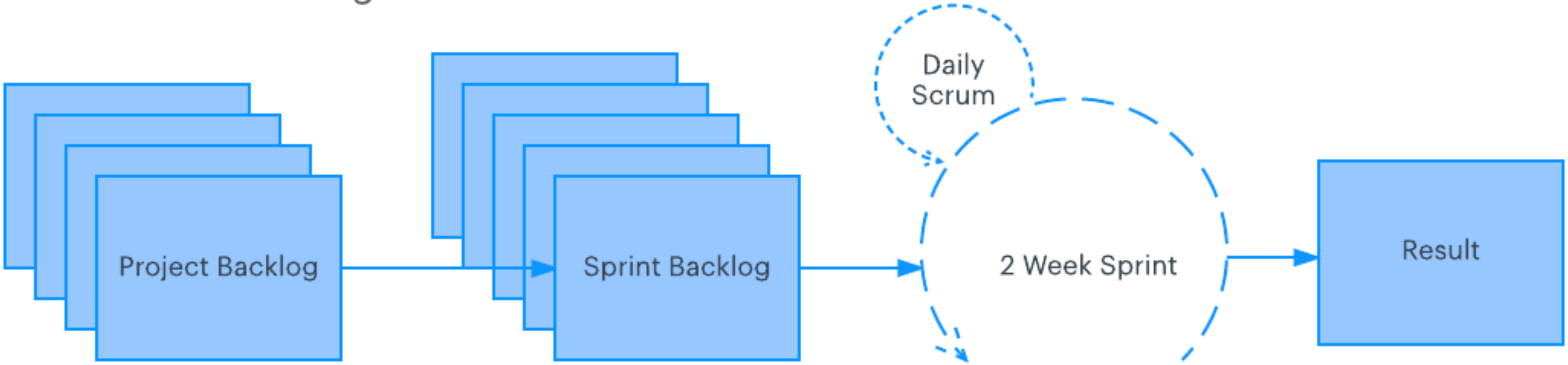
1. Our highest priority is to satisfy the customer through early and continuous delivery of valuable software.
2. Welcome changing requirements, even late in development. Agile processes harness change for the customer's competitive advantage.
3. ~~Deliver working software frequently, from a couple of weeks to a couple of months, with a preference to the shorter timescale.~~
4. ~~Business people and developers must work together daily throughout the project.~~
5. Build projects around motivated individuals. Give them the environment and support they need, and trust them to get the job done.
6. ~~The most efficient and effective method of conveying information to and within a development team is face-to-face conversation.~~
7. ~~Working software is the primary measure of progress.~~
8. Agile processes promote sustainable development. The sponsors, developers, and users should be able to maintain a constant pace indefinitely.
9. Continuous attention to technical excellence and good design enhances agility.
10. ~~Simplicity--the art of maximizing the amount of work not done--is essential.~~
11. The best architectures, requirements, and designs emerge from self-organizing teams.
12. ~~At regular intervals, the team reflects on how to become more effective, then tunes and adjusts its behavior accordingly.~~

SCRUM

- Sprints (typically two weeks)
  - Planned in advance, executed, then reviewed at the end in a retrospective
- 15-minute daily standup
  - What did you work on yesterday
  - What will you work on today
  - Are there any roadblocks preventing you from proceeding?
- Three main focuses
  - Team works collaboratively and stays in sync
  - Shippable deliverable at the end of each sprint
  - Continuous improvement (learn from retrospective, apply in planning session)

SCRUM

Scrum Process Diagram

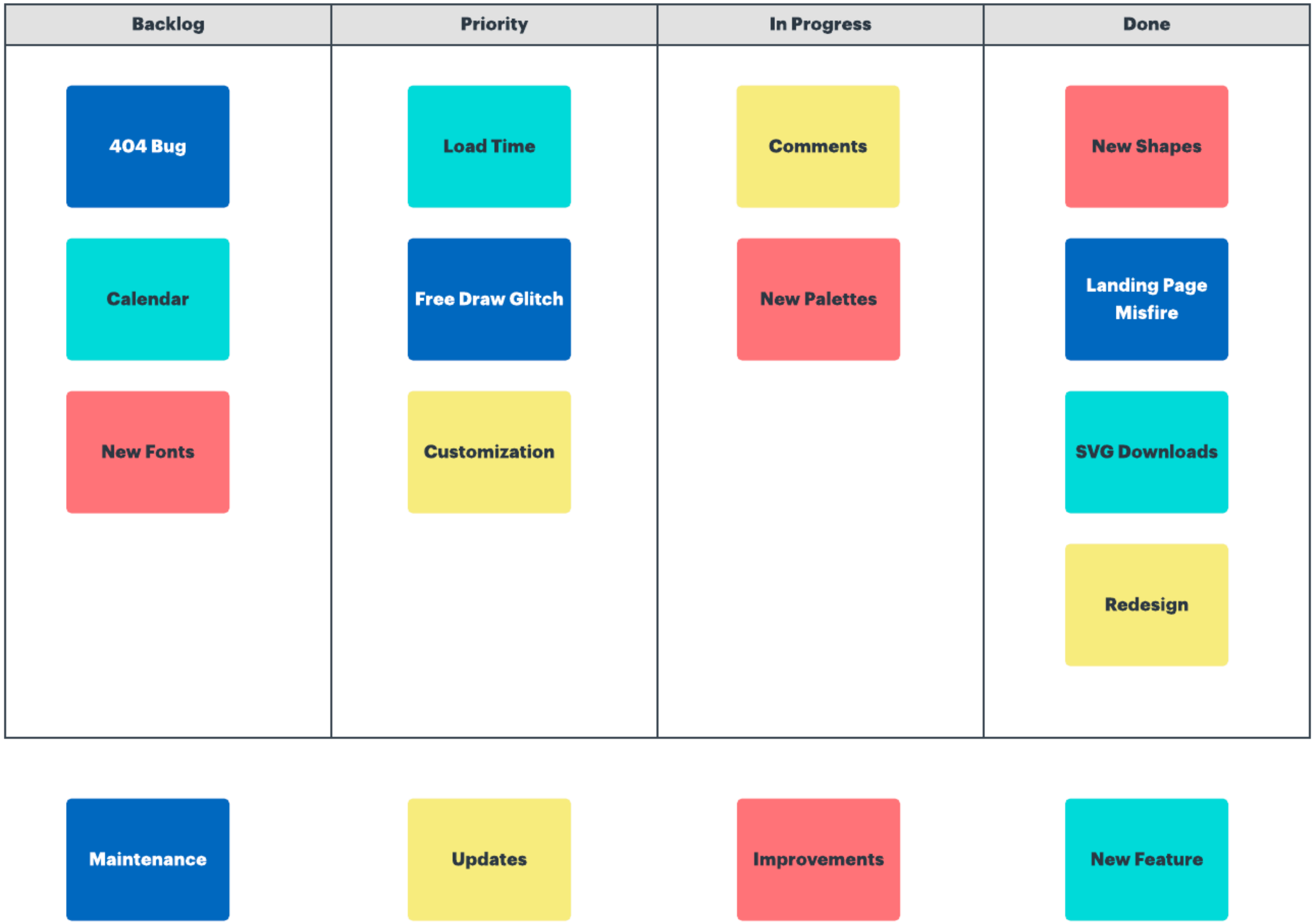


KANBAN

- Kanban board
- Strict Work-In-Progress limits in each lane
  - When a limit is reached, no new work can enter the column ('active development', 'QA', 'UAT', etc.)
  - Helps team identify bottlenecks
- Continuous improvement: team meets regularly to discuss changes that need to be made, correct, and adjust
- Regular changes, but no sudden or dramatic changes

KANBAN

Kanban Board



Backlog	Priority	In Progress	Done
404 Bug	Load Time	Comments	New Shapes
Calendar	Free Draw Glitch	New Palettes	Landing Page Misfire
New Fonts	Customization		SVG Downloads
			Redesign

Maintenance

Updates

Improvements

New Feature

RECOMMENDATIONS

- Example: Scrum for larger projects
  - Reasons go here
- Example: Hire a project manager

PART 2

PRODUCT MANAGEMENT

PRODUCT MANAGEMENT AT EXAMPLE

- Description of current state of Product Management

STRENGTHS

- Example: Product is always operating in alignment with business strategy

WEAKNESSES

- Example: Requirements and goals are often unclear

PRODUCT MANAGEMENT AT EXAMPLE: THE FUTURE

- Description of how the role of Product Management may evolve as the company evolves

PROPOSED FUTURE STATE FOR PRODUCT MANAGEMENT

Pretty diagram should go here

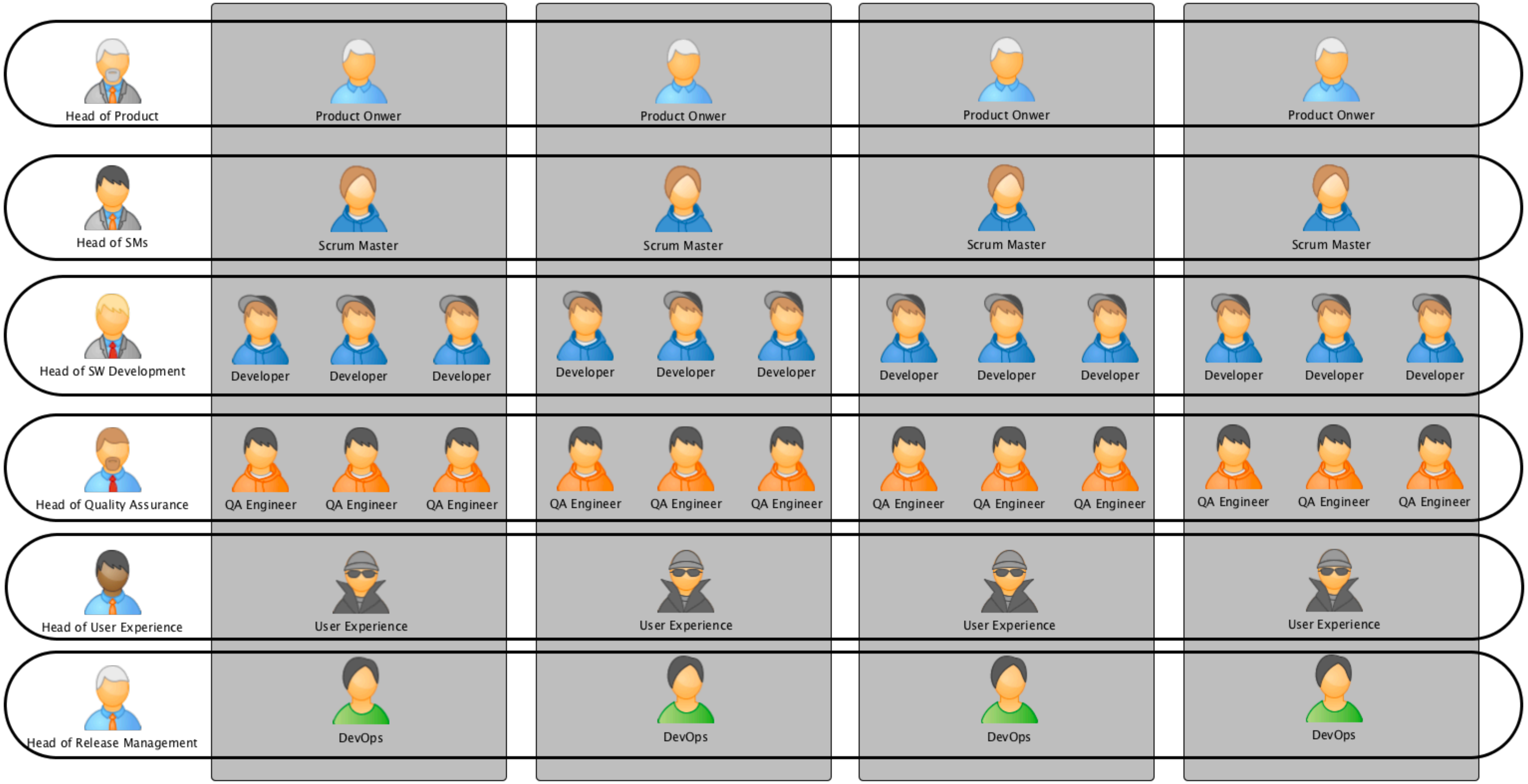
BENEFITS OF CROSS FUNCTIONAL TEAMS

- High level of expertise in product or functional area
- Team bonding
- Team trust
- Better understanding of goals and roadmap
- Facilitates quick communication and problem resolution
- Easy to scale or pivot as business needs change
- Clarity of business impact and value by team

PRODUCT MANAGERS SHOULD BE EMBEDDED IN CROSS FUNCTIONAL TEAMS

- Description of how that is different from the current state

EXAMPLE CROSS FUNCTIONAL STRUCTURE



Head of Product	Product Onwer	Product Onwer	Product Onwer	Product Onwer
Head of SMs	Scrum Master	Scrum Master	Scrum Master	Scrum Master
Head of SW Development	Developer Developer Developer	Developer Developer Developer	Developer Developer Developer	Developer Developer Developer
Head of Quality Assurance	QA Engineer QA Engineer QA Engineer	QA Engineer QA Engineer QA Engineer	QA Engineer QA Engineer QA Engineer	QA Engineer QA Engineer QA Engineer
Head of User Experience	User Experience	User Experience	User Experience	User Experience
Head of Release Management	DevOps	DevOps	DevOps	DevOps

PART 3

ALIGNMENT AND ENGAGEMENT

ALIGNMENT AND ENGAGEMENT

- The ability to build and deliver software solutions efficiently and ship them regularly is a means to an end
- Our raison d'être is to create value and grow the business
- It is important to have a system for ensuring the alignment of development work with business value
- That system should also help employees to feel engaged with broader business goals in order to drive better decision making

ALIGNMENT AND ENGAGEMENT

- Current state of alignment and engagement

RECOMMENDATIONS

- Example: Adopt the OKR framework for company and team goal setting

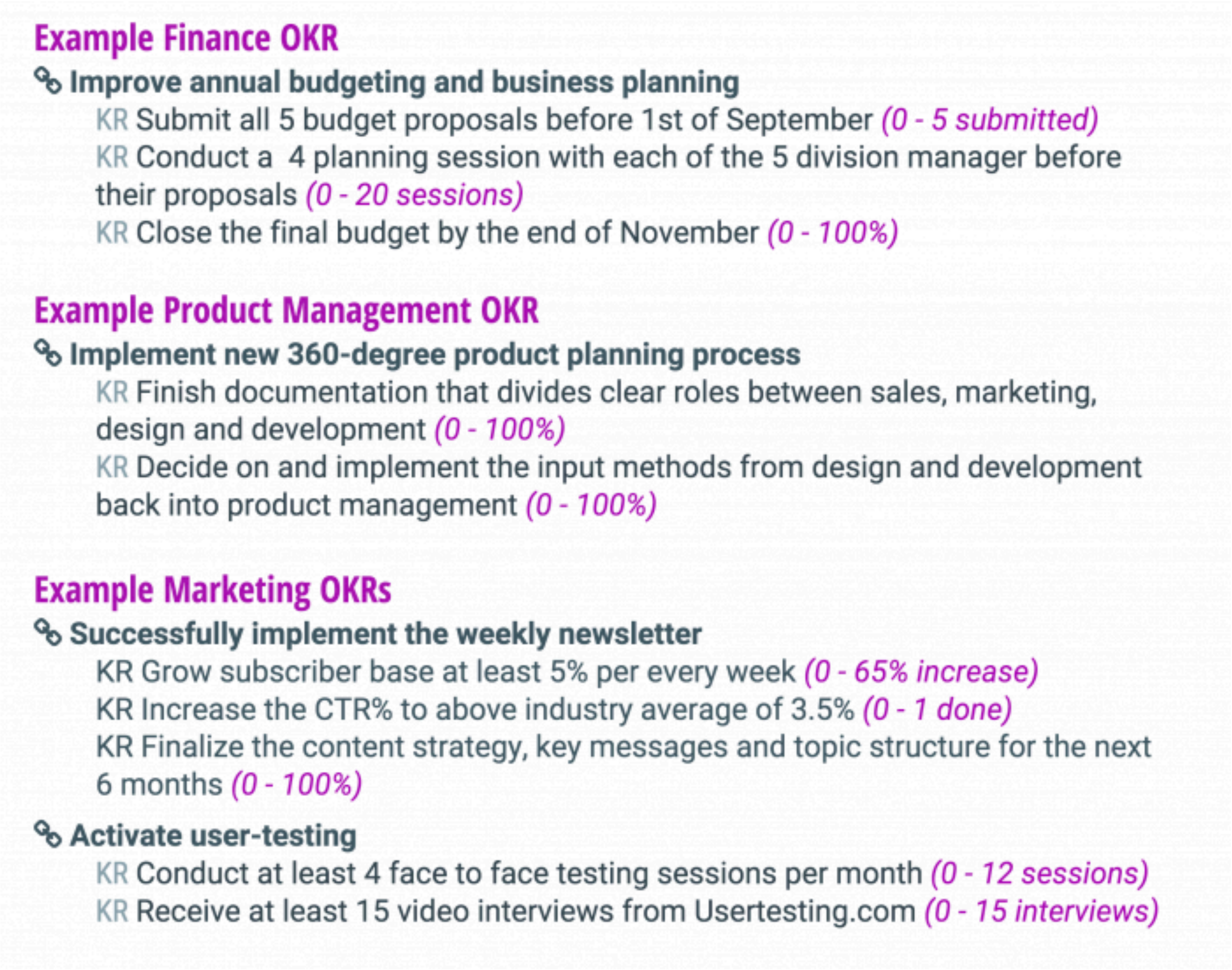
OKR – OBJECTIVES AND KEY RESULTS

- Very simple, low overhead system
  - Create 3 or 4 Goals on a quarterly or annual basis
  - Beneath them, some Key Results that, once achieved, achieve the goal
  - There can be company OKRs, team OKRs, and/or personal OKRs

OKR – OBJECTIVES AND KEY RESULTS

- First created at Intel, now used at Google, LinkedIn, Uber, and many small companies
- No custom system needed: best managed via spreadsheet (even at Google)
- "OKRs have helped lead us to 10x growth, many times over. They’ve helped make our crazily bold mission of 'organizing the world’s information' perhaps even achievable. They’ve kept me and the rest of the company on time and on track when it mattered the most." - Larry Page

OKR – EXAMPLES



Example Finance OKR

Improve annual budgeting and business planning

KR Submit all 5 budget proposals before 1st of September *(0 - 5 submitted)*

KR Conduct a 4 planning session with each of the 5 division manager before their proposals *(0 - 20 sessions)*

KR Close the final budget by the end of November *(0 - 100%)*

Example Product Management OKR

Implement new 360-degree product planning process

KR Finish documentation that divides clear roles between sales, marketing, design and development *(0 - 100%)*

KR Decide on and implement the input methods from design and development back into product management *(0 - 100%)*

Example Marketing OKRs

Successfully implement the weekly newsletter

KR Grow subscriber base at least 5% per every week *(0 - 65% increase)*

KR Increase the CTR% to above industry average of 3.5% *(0 - 1 done)*

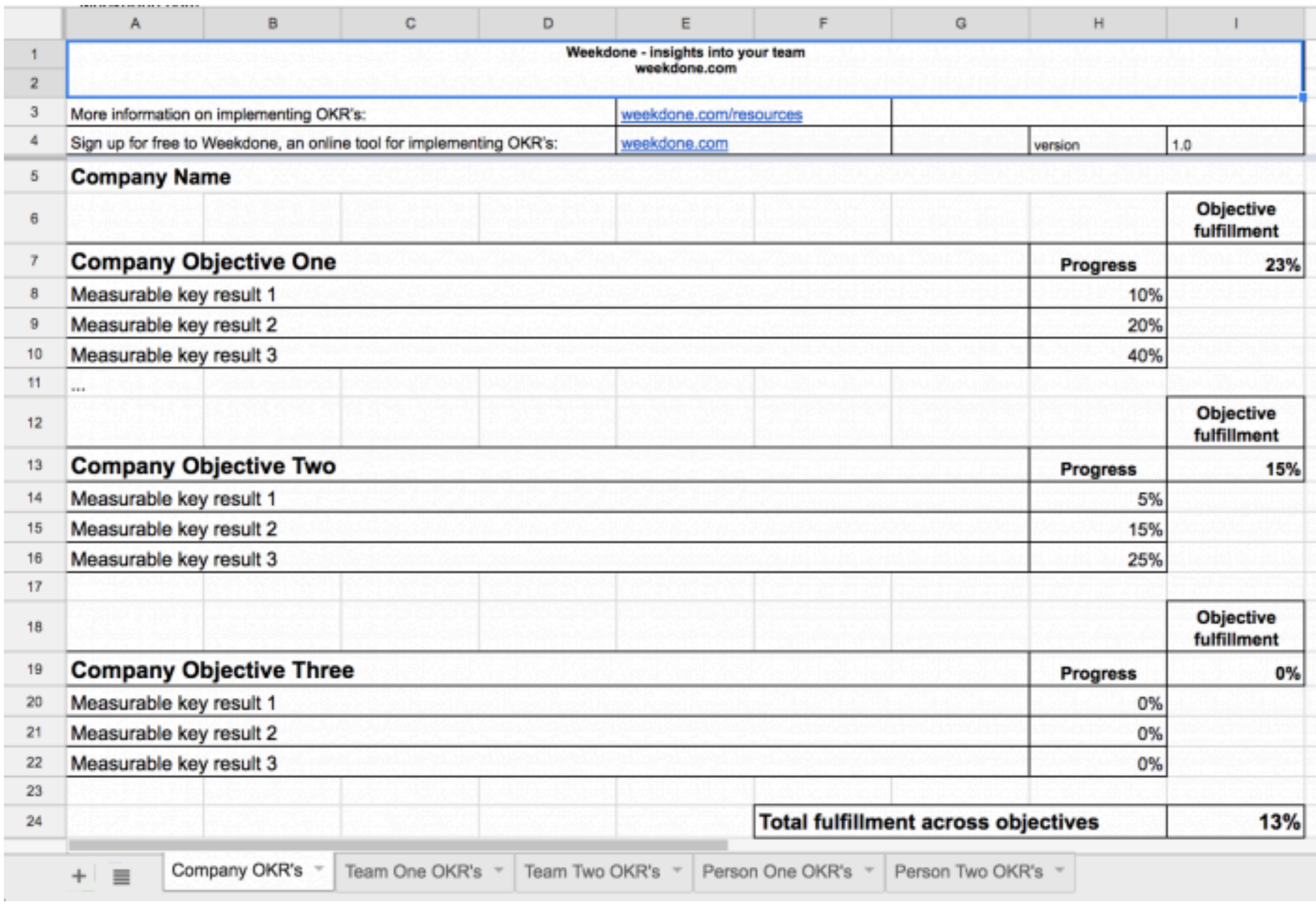
KR Finalize the content strategy, key messages and topic structure for the next 6 months *(0 - 100%)*

Activate user-testing

KR Conduct at least 4 face to face testing sessions per month *(0 - 12 sessions)*

KR Receive at least 15 video interviews from Usertesting.com *(0 - 15 interviews)*

OKR – SPREADSHEET FORMAT EXAMPLE



	A	B	C	D	E	F	G	H	I
1	Weekdone - insights into your team weekdone.com								
2									
3	More information on implementing OKR's:				weekdone.com/resources				
4	Sign up for free to Weekdone, an online tool for implementing OKR's:				weekdone.com			version	1.0
5	Company Name								
6									Objective fulfillment
7	Company Objective One							Progress	23%
8	Measurable key result 1							10%	
9	Measurable key result 2							20%	
10	Measurable key result 3							40%	
11	...								
12									Objective fulfillment
13	Company Objective Two							Progress	15%
14	Measurable key result 1							5%	
15	Measurable key result 2							15%	
16	Measurable key result 3							25%	
17									
18									Objective fulfillment
19	Company Objective Three							Progress	0%
20	Measurable key result 1							0%	
21	Measurable key result 2							0%	
22	Measurable key result 3							0%	
23									
24						Total fulfillment across objectives			13%

OKR – FURTHER READING

- Measure What Matters by John Doerr (Google)
- Radical Focus by Chistina Wodtke (short read)

PART 4

SOFTWARE DEVELOPMENT

CURRENT PROCESSES

- Description of current processes

STRENGTHS

- Example: There are few hurdles preventing rapid development
- Example: Source code management is used effectively

WEAKNESSES

- Example: Deployments are not automated
- Example: Difficult to track ongoing work, changes, progress

OPPORTUNITIES

- Example: Many of the right tools are in place and can be leveraged more effectively

PROBLEMS AND SOLUTIONS

- Example: Deploys are not automated
- Example: Code reviews should be better leveraged

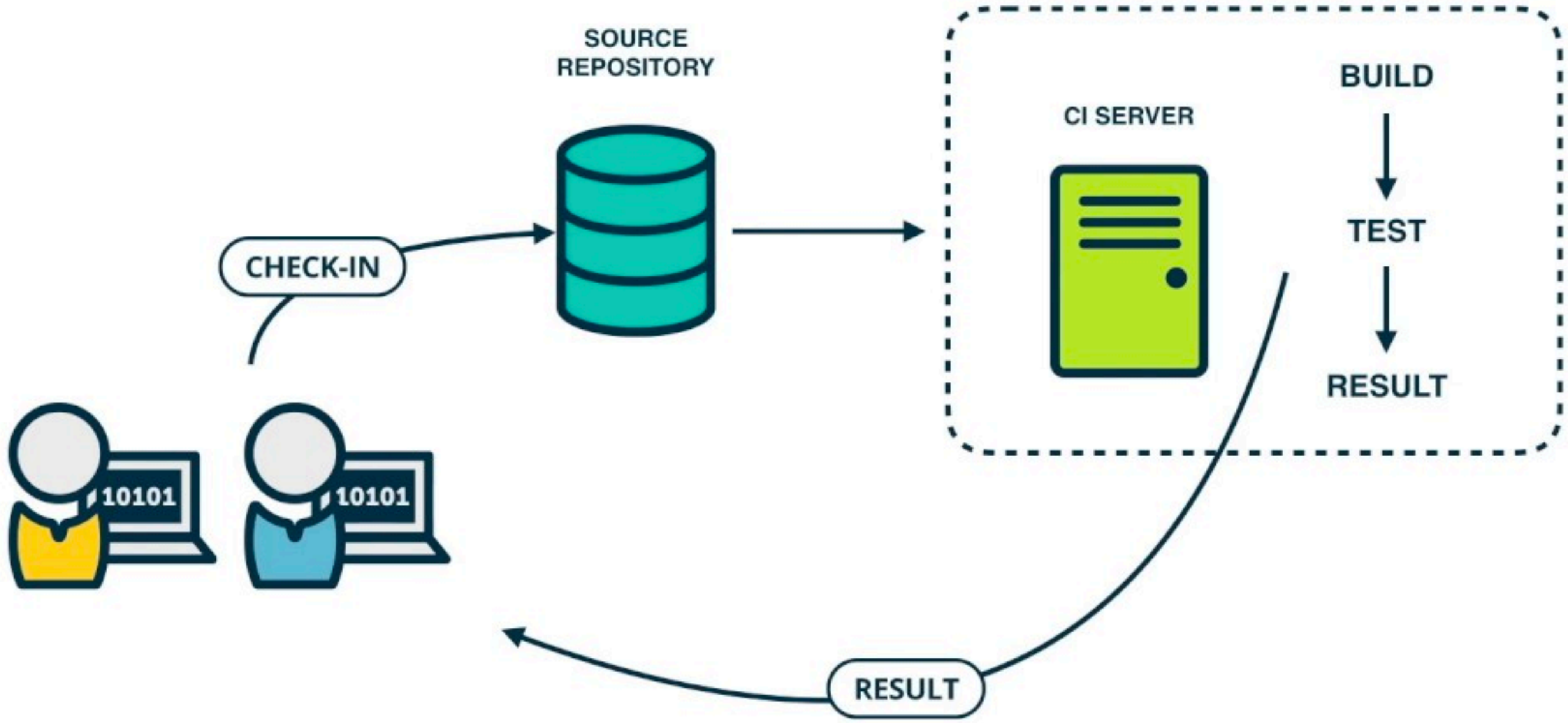
EXAMPLE PROBLEM: DEPLOYS ARE NOT AUTOMATED

- Currently deploying code is a manual process
- If key team members are not available, code cannot be released
- There is not always a clear process for rolling back releases in case of problems
- There is no history of what was released, and when, increasing the challenge of debugging issues

EXAMPLE SOLUTION: CONTINUOUS INTEGRATION

- Development practice that requires developers to integrate code into a shared repository several times a day.
- Each check-in is then verified by an automated build, allowing teams to detect problems early.
- Many systems for managing this, both hosted and SaaS

Continuous Integration (CI)



EXAMPLE SOLUTION: CONTINUOUS INTEGRATION

- Proposal for implementing the change goes here

EXAMPLE PROBLEM: CODE REVIEWS SHOULD BE BETTER LEVERAGED

- Description of the current code review process, if any

CODE REVIEWS SERVE MULTIPLE PURPOSES

- Better code quality
- Finding bugs
- Learning/knowledge transfer
- Increase sense of mutual responsibility
- Finding better solutions

EXAMPLE SOLUTION: CODE REVIEWS SHOULD BE PART OF A DEFINED PROCESS

- Code reviews should be mandatory for:
  - Critical systems (define this somewhere)
  - Changes with security impact (define this somewhere)
  - Junior engineers (define this somewhere)
- Code reviews should be suggested for:
  - Code that will be used by multiple teams (e.g. APIs)
- Code reviews should be optional for:
  - Everything else

OPEN QUESTIONS

- List of open questions goes here, with followup steps

PART 5

SYSTEMS AND INFRASTRUCTURE

SYSTEMS AND INFRASTRUCTURE

Pretty systems architecture diagrams go here

If you don't have them, creating them collaboratively with the team is a great way for everyone to learn

SYSTEMS AND INFRASTRUCTURE

- Systems and infrastructure concerns or other key points go here

CLASSIFYING SYSTEMS

- Help your audience understand how your systems are tied to the business
- Example: Customer Facing
- Example: Employee Facing

CLASSIFYING SYSTEMS

- **Customer Facing**
  - Example system 1
  - Example system 2
- **Employee Facing**
  - Example system 1
  - Example system 2

RECOMMENDATIONS

- Example: We should have different standards of responsiveness, redundancy, and availability by system
  - These standards should be explicitly defined and met

EXAMPLE RECOMMENDATION DETAILS

- Customer Facing
  - Deliver content and become interactive in under 5 seconds
  - Respond to user interactions within 50ms
  - 99.999% uptime during U.S. business hours (12:00pm-1:00am UTC): down less than 5.26 minutes per year
- Employee Facing
  - Deliver content and become interactive in under 10 seconds
  - 99.9% uptime during U.S. east coast business hours (8:00am-6:00pm EST): down less than 8.77 hours per year

DISASTER RECOVERY

- What systems are in place to mitigate a disaster (e.g. sustained power or network outage to data centers, DDoS, etc.)?
- Description of the current state of Disaster Recovery

DISASTER RECOVERY: RECOMMENDATIONS

- Example: Company should develop a Disaster Recovery plan
  - The plan should cover each product, production systems, and office systems / business continuity

PROBLEMS AND SOLUTIONS

- Deeper dives into key problems, along with proposed solutions, can go here

PART 6

TOOLS

TOOL ADOPTION

Charts, diagrams, or tables should go here

RECOMMENDATIONS

- Example: There should be a single owner of all tool decisions (CTO)
- Example: Tool decisions should be based primarily on usability

PART 7

SECURITY

STATE OF SECURITY AT COMPANY

- Description of the current state of security goes here

POSITIVES

- Example: Multi-factor authentication is used for most important systems

NEGATIVES

- Example: No processes are in place to regularly test for vulnerabilities
- Example: There is limited protection against phishing attempts

RECOMMENDATIONS

- Example: A security professional or firm should be engaged to assess critical systems

PART 8

ORG STRUCTURE

TECHNOLOGY GROUP

- Description of current de facto org structure goes here, along with areas of focus and connection to different areas of the business where relevant

TEAM STRUCTURE

- Assessment of current team structure goes here

POSSIBLE FUTURE TEAM STRUCTURE

- Description of possible future team structure goes here if necessary, along with impact, pros, and cons

PART 9

PEOPLE

ROLES

- Description of current roles and gaps, as well as evaluations of people where possible

PERSONNEL RISK

- Description of key personnel whose loss would have an immediate negative impact on the business
- Plans for mitigating this risk

RECOMMENDATIONS

- Example: Take steps to decentralize knowledge and responsibility

GAPS

- Example: Several critical roles are currently not being performed within the technology and product groups
    - Deeper dive into missing roles and solutions
- Example: Management structure is defined but suboptimal
    - Deeper dive into problems and solutions

RECOMMENDED INCREMENTAL HIRES

- Description of recommended incremental hires goes here

RECOMMENDED ROLE CHANGES

- Description of recommended role changes goes here, including terminations if necessary

PART 10

THE CTO ROLE

THE VALUE OF THE CTO ROLE

- The CTO role spans all of the areas covered so far and must examine all those areas in consideration of both short and long term organizational needs
- The aim is for the person in this role to ensure that there is:
  - A healthy and functional organizational structure
  - Good, highly skilled, engaged people in that structure
  - A clear and accurate strategy for those people to operate against
  - The translation of that strategy into deliverable, measurable goals
  - Adequate processes to facilitate the timely and satisfactory execution of those goals with as little friction as possible
  - Adequate systems and tools with which to operate
  - Strong, resilient technology systems to reliably operate the business

THE CTO AT COMPANY SHOULD BE

- Description of the CTO role at Company

THE CTO AT COMPANY SHOULD NOT BE

- Description of areas where the CTO needs to partner, delegate, or eliminate

CRITICAL QUESTIONS

- Any remaining critical questions go here.

END OF ASSESSMENT

THANK YOU!