people → values → agility → practices
roles

product owner
- product direction (what we build)
- snapshots (roadmaps)

engineering lead
- how we build it
- technical strategy
- team member management

teammember
- how long it takes
- contributes to team success

© 2019 James Newkirk
company values

communication
we have an obligation to communicate.

integrity
we work with customers and prospects openly, honestly, and sincerely.

respect
we treat others as we would like to be treated.

efficiency
we are satisfied with nothing less than the very best in everything we do.
recognize and reward only those individuals who's actions demonstrate the values

**my values**

- **transparency**
  - tell the whole truth, without delay

- **minimalism**
  - extreme simplicity

- **guidance**
  - clear, specific, and sincere

- **sustainability**
  - meet the needs of the present without sacrificing the future
values bring purpose to practices

practices bring accountability to values
an example

what is this driving technique?

why do this?

is it correct?
participate

your values  
3-4 things that matter to you

company values  
company stated values

are they the actual values? If not, where do they differ?

shared values

you

company

team
choose your approach

predicting the future
the planning fallacy

definition
predictions about how much time will be needed to complete a future task display an optimism bias and underestimate the time needed

example
37 psychology students were asked to estimate how long it would take to finish their senior thesis
average : 33.9 days
optimistic : 27.4 days
pessimistic : 48.6 days

results
average completion was 55.5 days
30% completed in the time predicted

Daniel Kahneman and Amos Tversky
Roger Buehler

finish

the hardest thing to do is finish
unrealistic goals discourage success
work in progress is the enemy
Cut your goals in half and finish practices.
progressive refinement

minimum viable product
<table>
<thead>
<tr>
<th><strong>minimum viable product</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Wrong</strong></td>
</tr>
<tr>
<td>the crappiest product you could possibly release</td>
</tr>
<tr>
<td></td>
</tr>
</tbody>
</table>

Identify Core Value
layers

layers

5%
layers 20%

layers 40%
progressive refinement metaphor

an algorithm that quickly reveals course structure and gradually reveals additional detail over time
progressive refinement
20%

progressive refinement
40%
maintain
continuity

roadmaps
snapshots

snapshot practice

now
these are items that you expect to deliver in the next 3 months.
the commitment to the customer is that if something changes you have to communicate immediately.

next
these are items that you expect to be able to deliver in the next 4-6 months.
the list is sorted in priority order.

under consideration
these are items that you are considering for delivery beyond 6 months.
as is the case with “next” these are items in priority order.
<table>
<thead>
<tr>
<th>prioritization</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>One - # 1</strong></td>
</tr>
<tr>
<td>there can only be one #1 priority.</td>
</tr>
<tr>
<td>subsequent items need to be in descending order of priority</td>
</tr>
</tbody>
</table>

---

**defects**
defect rate

defect discovery
need to understand the rate in which you are discovering defects and correcting them

trends
over time is the team getting better, fewer new defects, better at fixing them when found

defect practice

now
customers are losing time, money, or promised value
stop what you are doing and fix it!
“the system is not maintaining the state between our system and the external payment system resulting in erroneous billing”

next
the software is not performing as expected, but customers are able to receive the value they expect
finish the current sprint and then correct it in the next sprint

never
let’s stop kidding ourselves, these will likely never be fixed
close: Won’t fix
if you don’t close these they become a huge tax on your development
just-in-time

use the aging report to indicate which defects should be closed due to how long they have been in the database.

aging

use the aging report to indicate which defects should be closed due to how long they have been in the database.

defect estimation

inaccurate estimating the size of a defect can be very difficult

story points?

story points are tied to delivering value

impede progress

if the team works a significant amount of time on defects then the overall velocity will decline

you get points for value delivered, not work done
The product owner prioritizes all defects independent of stories at the start of the sprint. When the sprint begins, the product owner reviews the work completed, the incoming defects, and any other changes to the list of defects. The product owner then decides the number of people they want to dedicate to fixing defects in the iteration. The product owner reprioritizes the list based on the current needs and resources.
SWAT Team

<table>
<thead>
<tr>
<th>Date</th>
<th>SWAT</th>
<th>Development</th>
</tr>
</thead>
<tbody>
<tr>
<td>05.16</td>
<td>2</td>
<td>5</td>
</tr>
<tr>
<td>05.30</td>
<td>2</td>
<td>5</td>
</tr>
<tr>
<td>06.13</td>
<td>2</td>
<td>5</td>
</tr>
<tr>
<td>06.27</td>
<td>2</td>
<td>5</td>
</tr>
<tr>
<td>07.11</td>
<td>6</td>
<td>6</td>
</tr>
<tr>
<td>07.25</td>
<td>7</td>
<td>6</td>
</tr>
</tbody>
</table>
exploring words

estimate
guess

educated guess
crowd-sourced educated guess

an example

pages: 694
audio: 33 hours and 46 minutes
read the first chapter, record the time, and then estimate
user story points

story points
a value that depicts the overall effort that will be required to fully implement a user story from the product backlog

point values
assign a point value to each item
relative value is what matters

must include
the amount of work to do
the complexity of the work
any risk or uncertainty in doing the work

planning poker practice

definition
an agile estimating technique that is consensus based
cards have the following values, ½, 1, 2, 3, 5, 8, 13, 20, 40, 100, ?

practice
the estimators discuss the user story with the product owner
each estimator selects one card to represent the estimate.
all cards are revealed at the same time
if all estimate are the same that becomes the estimate
if not, there is a discussion
once the discussion completes each estimator reselects a card and then they are revealed

Mike Cohn, https://www.mountaingoatsoftware.com/blog/what-are-story-points

Mike Cohn, https://www.mountaingoatsoftware.com/agsile/planning-poker
**crumb scale practice**

**size ordering**
- start from an unordered list of stories
- one-by-one ask the question, "is this bigger than that?"

**point distribution**
- points: ½, 1, 2, 3, 5, 8, 13, 20, 40, 100
- start at the bottom of the list, give the value ½ to the stories that are smallest
- when you find a story that is twice as big as the ½ stories switch to the value of 1
- "is this twice as big as that?"

---

**key performance indicators (kpi)**
**kpi practice**

realistic

customer focused (focused on outcomes)

actionable (why does it matter?)

understand precision and present it accordingly

**kpi example**

“zero customer impacting issues related to our deployments”

why - having no customer impacting issues when we deploy will increase our customers confidence in the overall solution and reduce the anxiety of the team members deploying the software

how - this will be measured by counting the number of hot fixes that are needed within the first 48 hours of every deployment

who - engineering leader is responsible for this metric

the metric will be reviewed at each monthly operations review
making code ready for the next challenge

developer practices

- test-driven development
  - test first programming

- continuous integration
  - keep the code deployable all the time

- refactoring
  - clean as you go

- collective ownership
  - no silos
focus on outcome
minimize work in progress
succeed together

what questions do you have?
i need your help!

thank you