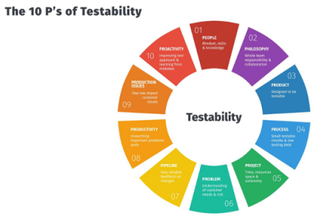

TESTABILITY:
People - mindset, skills, knowledge
Philosophy - collaboration
Product - designed for testability
Process - small chunks, low debt
Project - time, resources, autonomy
Problem - understanding risk
Pipeline - fast, reliable change feedback
Productivity - unearthing problems early
Production Issues - low-impact customer issues
Proactivity - improving approach, learning from mistakes

1. Place the green testability post-it notes into the categories, you believe they belong to.
2. In a different colour to green/yellow, add your own questions to each testability category.
3. Place the yellow automatability post-it notes into the categories, you believe they belong to.
4. In a different colour to green/yellow (and your chosen testability), add your own question to each automatability category.

Return to Day 1 activity home page	People	Philosophy	Product	Process	Project	Problem	Pipeline	Productivity	Production issues	Proactivity
Testability	What testing skills do the team have? Who has all the product knowledge? Is the team actively encourage to experiment?	Does the whole team take responsibility for testing? Test early in development process Are the developers encouraged to do exploratory testing?	What risks are currently the most important? Can tester access information or supply information	Are stories adequately broken down? Who is involved in story planning? How many known bugs are currently in production? Testability brought up in grooming	Does the project have a deadline? Is there time allotted to fix issues that come up from testing?	Do you know who your customers are? Do we have a good understanding of dependencies?	How long does a build currently take? Reproduction of client issus How much testing is done as early as possible? What are the steps fo the currently CI system?	How easy is it to create test data? What is the mean time to discovery for high impact bugs? How many test environments are there?	How do customers report issues in production?	How often does the team have retrospectives?
Automatability	What automation tools does the team have experience with? Who is going to be maintaining the automated checks? Automated Testing a part of DOD	Does the whole team contribute to automated checks? Is the team open to new tooling? Do the developers contribute to the quality of the automated check code?	How many APIs does the product have?	Are problems with the automation framework identified early? Is there a big backlog of things waiting to be automated?	Does the project have budget for test environments? Has the project estimated for time to set up an automation framework?	Does the UI have identifiers to make it easier to automate? Are there any known issues that would impact our automation efforts? Do we understand what risks need to be automated? Do we know what flows the customers value the most?	Does the existing pipeline contain any automated checks? How quickly are raised issues integrated into the backlog? Is there a pipeline already in place?	Are the automated checks ran as early as possible to provide rapid feedback? Are test environments reliably close to live environments? Does the automation efforts keep up each sprint?	Is there any monitoring in place for identifying potential production issues? Alerting and oncall	Does the team have regularly retrospectives on the automation efforts? RCA for Defects

Parts, features, protocols, layers, technology, users

Any spam protection?
If emails are sent, is there an opt out?
Is there an existing test framework we can leverage?
https://github.com/mwinteringham/restful-booker-platform/tree/trunk/end-to-end-tests
What programming language is the product written in?
JDK 15.0.2 Maven 3.6.3 Node 14.15.5 NPM 6.14.11
Are multiple locations in a franchise supported?
no just rooms
Can repeat customers make accounts?
Privacy
Language and availability of translations
Product owner involvement?

Are there any third party integrations?
What devices should be supported?
Is there user feedback after booking?
Are there data regulations we need to be concerned about (GDPR etc)?
Are there existing test accounts?
Are there Unit tests?
What is the technical architecture?
Support for accessibility (WCAG)/What level?

doesn't look like it, just libraries
not currently mobile optimized
yes, junit
couple of services hooked together with apis
currently 79 per lighthouse

How is data stored?
Is data viewable / queryable?
are there outstanding defects?
Does the product have an api?
Can the site be edited by the client?
How are bookings managed/canceled/modified after being made?
Can we get a demo?
Special characters
Access to log files?

dbs
yes through the APIs/sql
https://github.com/mwinteringham/restful-booker-platform/projects/1
Booking API http://localhost:3000/booking/swagger-ui/index.html
admin panel
admin panel
no support
http://localhost:3000/booking/actuator/logfile

Is there payment processing?
Will multiple languages be supported?
Is there a long-term roadmap we can access?
requirements around booking dates or costs?
Who may access the product?
Security
Performance expectations?
Can we test different parts (front end / api) separately?

https://github.com/mwinteringham/restful-booker-platform/issues
price under 999
public front end, "private" back end
needs fixing - admin/password
appears so