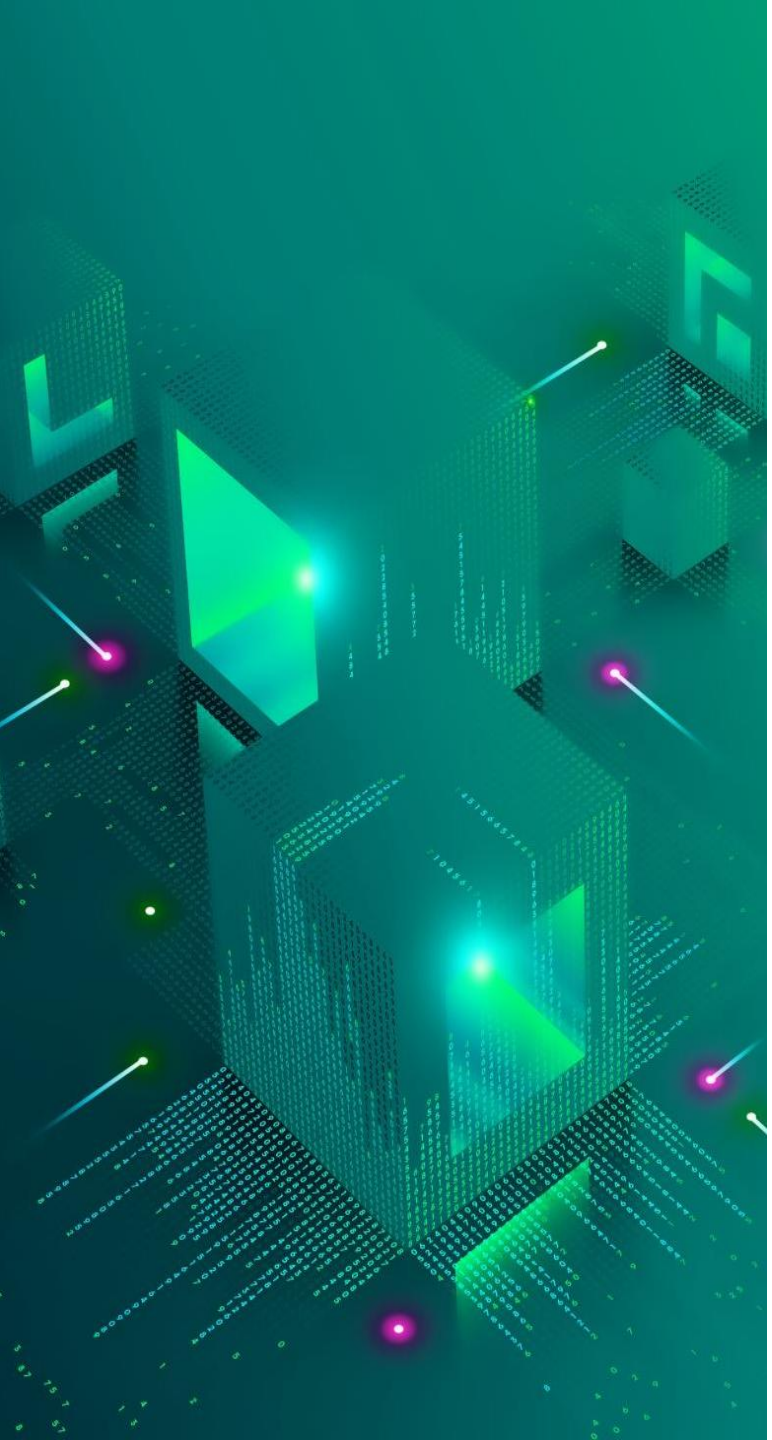


From Good to Great

USABILITY FOR SOFTWARE ENGINEERS

Dave Paquette
Principal Software Engineer - Microsoft

NORTHFIELD

improving
It's what we do. ™

online
business systems

CONQUEST

IcePanel

POLLARD
banknote limited

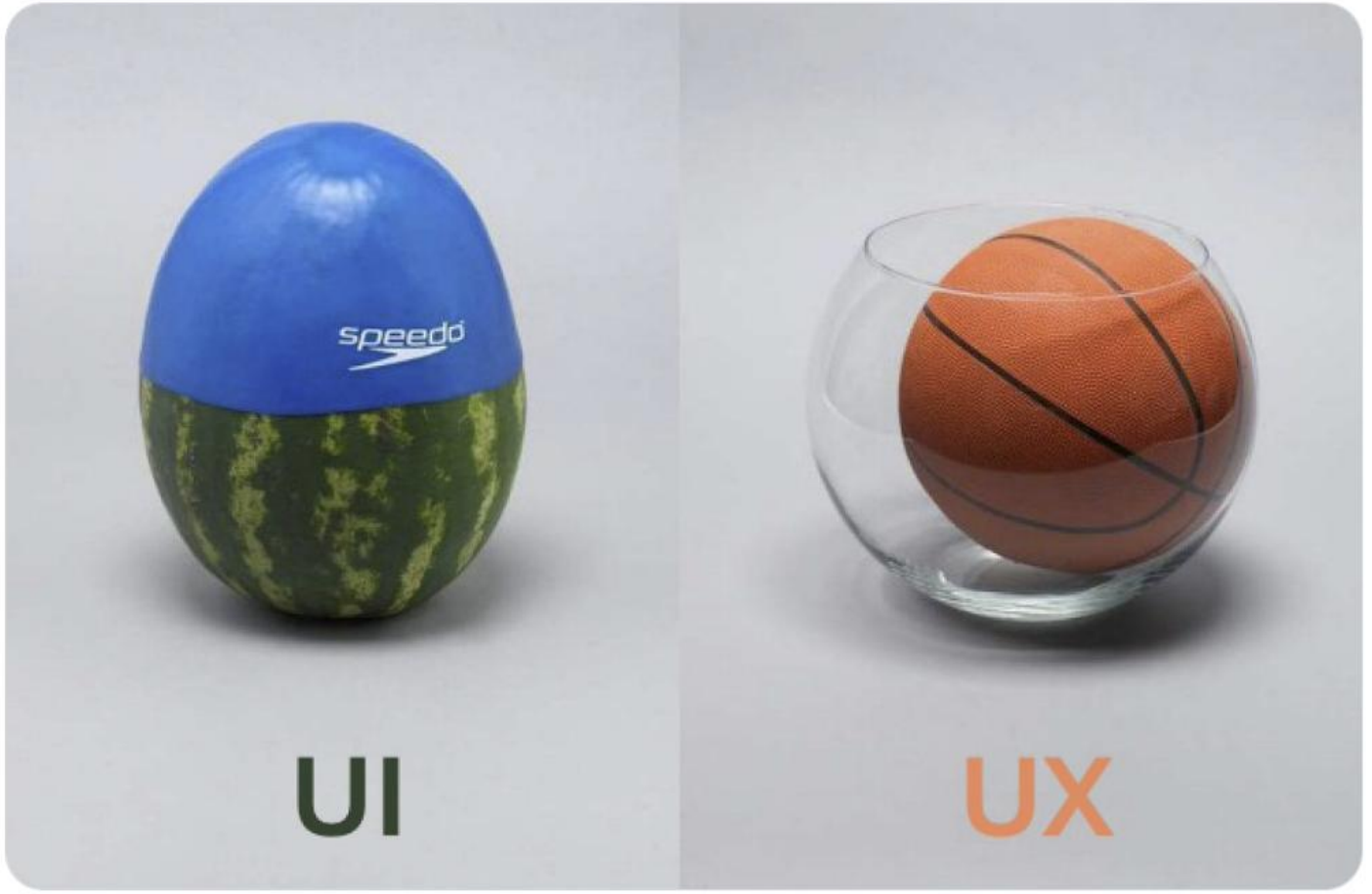
RICHARDSON

What is Usability?

User experience
Design

Sebastiaan de With
@sdw

I label increasingly nonsensical images with ‘UI’ and ‘UX’ and hope they get used in serious presentations



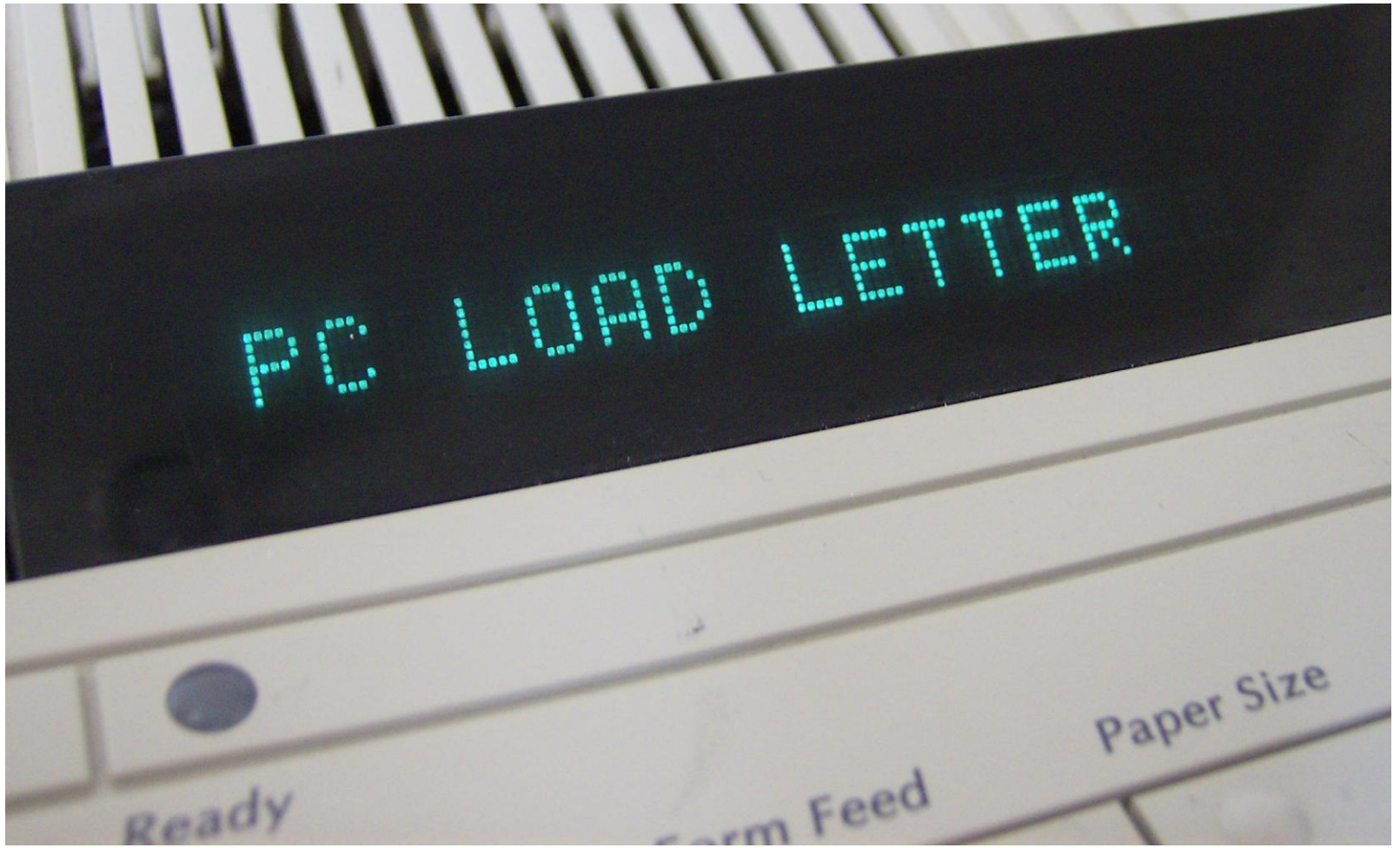
3:26 PM · Mar 15, 2016

How easy a user interface is to use

- 5 quality components
  - Learnability
  - Efficiency
  - Memorability
  - Errors
  - Satisfaction

- Utility = features you need
- Usability = Features are easy and pleasant to use

Useful = Usability + Utility



PC LOAD LETTER?

"WHAT THE !*#% DOES THAT MEAN?"

OFFICE SPACE (1999)

Small decisions can have a big impact on user satisfaction

How do we improve Usability?

- User testing
  - Test old designs before starting new design
  - Test competitor’s designs
  - Conduct field studies
  - Make paper / low fidelity prototypes of new design ides and test them
  - Iterate

TikTok
@ tired_actor

Discount Usability

WE CAN ALL CONTRIBUTE TO IMPROVING THE USABILITY

10 Usability Heuristics

10 general principles for interaction design

Broad rules of thumb and not specific usability guidelines

Jakob Nielsen (1994) https://www.nngroup.com/articles/ten-usability-heuristics/

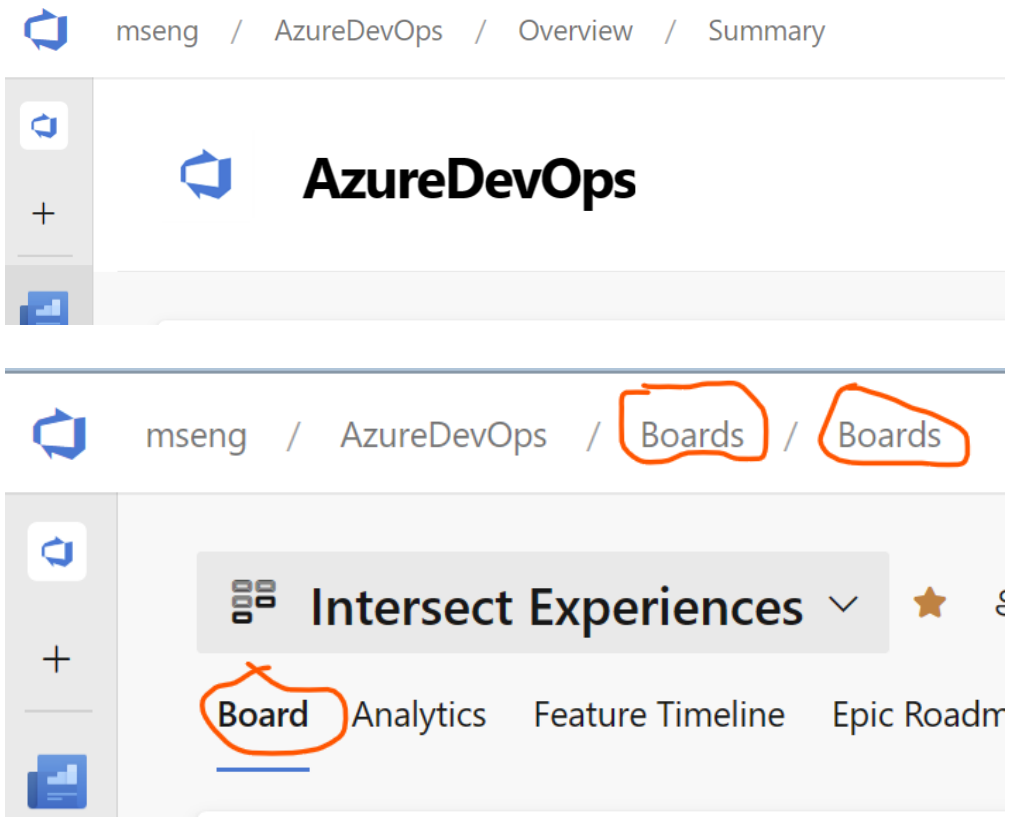
#1: Visibility of System Status

The design should always keep users informed about what is going on, through appropriate feedback within a reasonable amount of time.

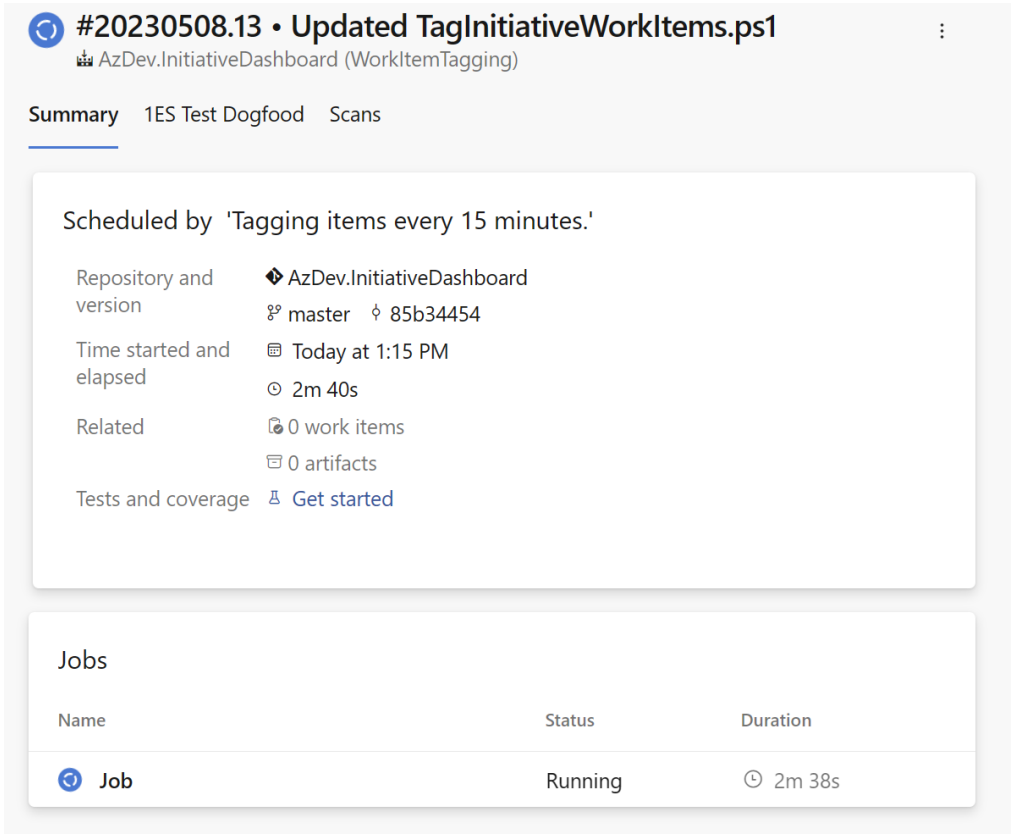
e.g. You are Here indicators on mall maps

#1: Visibility of System Status

Breadcrumbs



Pipeline Status



#1: Visibility of System Status

Response time (How users perceive delays)

- **< 0.1 second** - perceived as instantaneous
- **1 second** - user's flow of thought stays uninterrupted, but delay noticed
- **10 seconds** - limit for keeping user's attention focused on the dialog
- **> 10 seconds** - user will want to perform other tasks while waiting

#2: Match between system and the real world

- Speak the users' language
- Words, phrases, and concepts familiar to the user
- Follow real-world conventions, making information appear in a natural and logical order

e.g. When stovetop controls match the layout of heating elements, users can quickly understand which control maps to which heating element.

#2: Match between system and the real world

- Don’t assume your understanding of words or concepts will match those of your users

#3: User control and freedom

- Users often perform actions by mistake.
- Clearly marked "*emergency exit*" to leave the unwanted action

e.g. Digital spaces need quick emergency exits, just like physical spaces do.

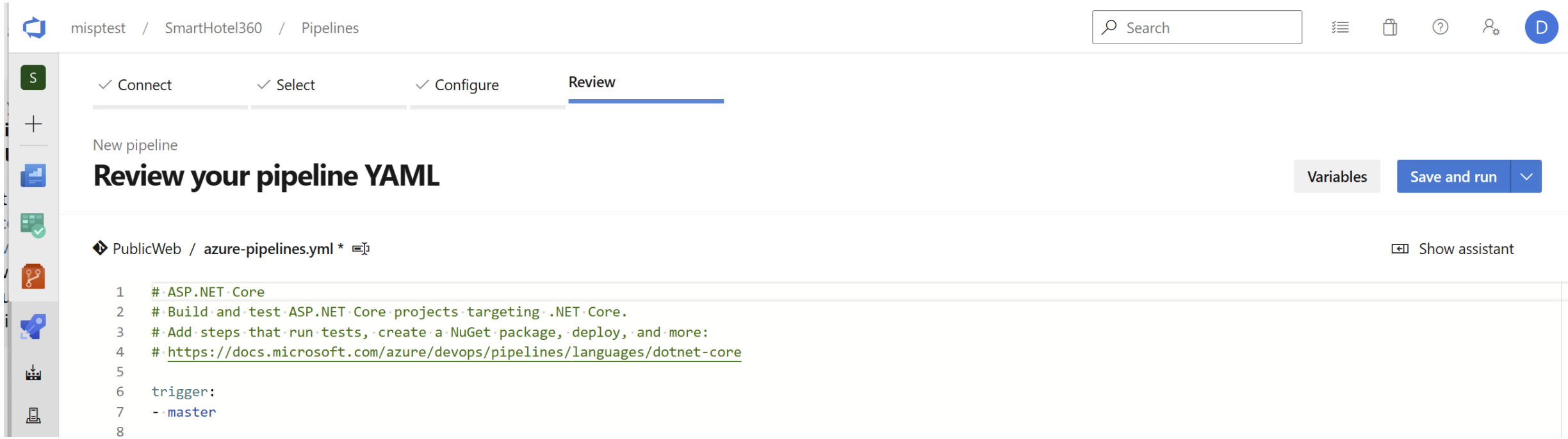
#3: User control and freedom

Users don’t like to feel trapped by the computer!

Strategies

- Cancel (allow the user to quit a task or multi-step process)
- Undo/Redo (can get back to previous state)
- Close (allow the user to close a new view)
- Defaults (for restoring a settings)

#3: User control and freedom

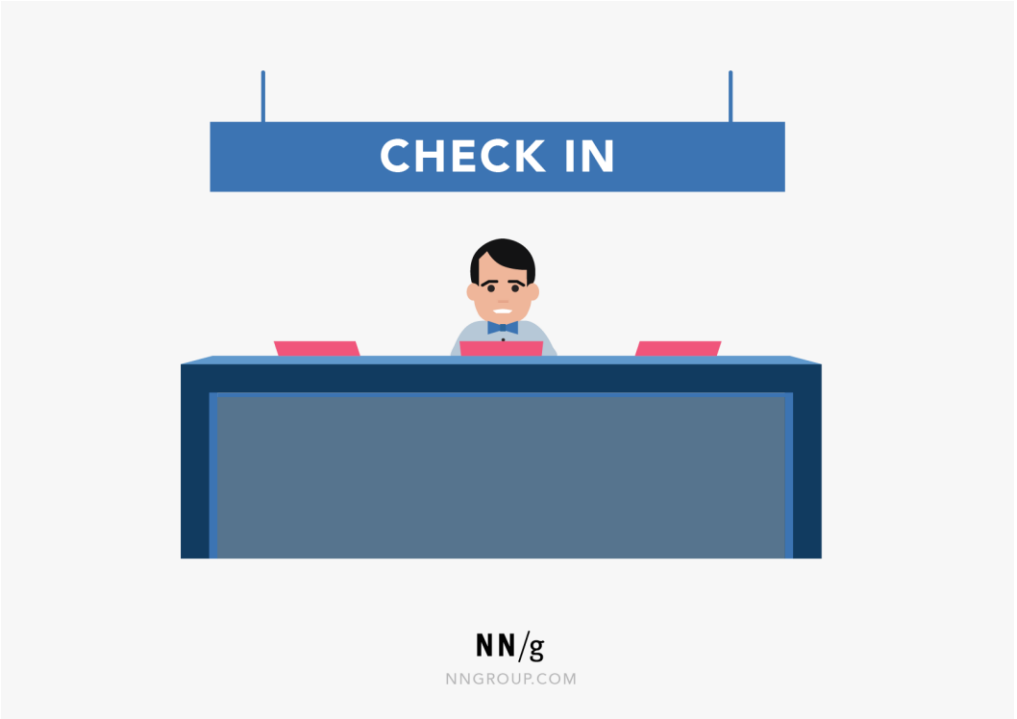


#4: Consistency and standards

- Users should not have to wonder whether different words, situations, or actions mean the same thing
- Follow platform and industry conventions

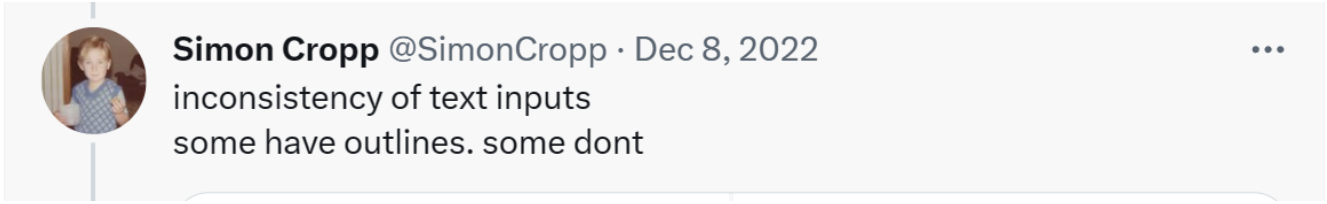
Ok Cancel Cancel Ok Done Never Mind Accept Dismiss

e.g. Checkin counters are usually located at the front of hotels. This consistency meets customers' expectations.

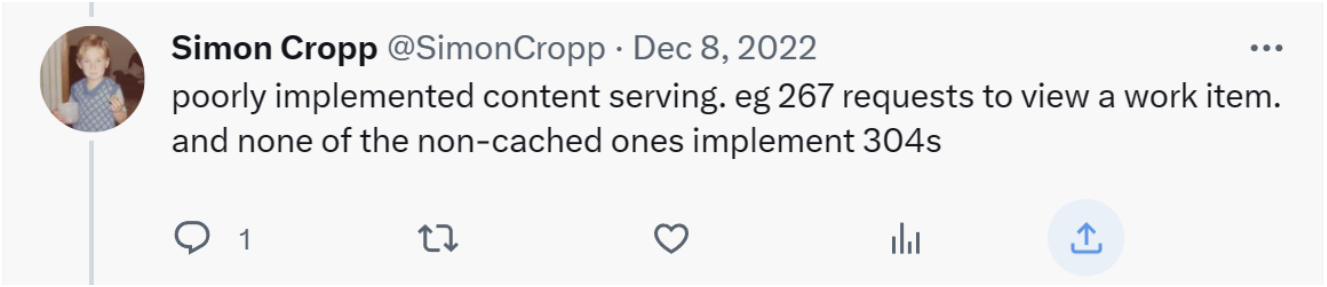


#4: Consistency and standards

- Internal Consistency
  - Use design system
  - Beware of naming consistency



- External Consistency
  - Recognizable icons
  - Don’t reinvent standards



#5: Error Prevention

- The best designs **prevent problems from occurring** in the first place
- Either eliminate error-prone conditions or check for them and present users with a confirmation option before they commit to the action

e.g. Guard rails on curvy mountain roads prevent drivers from falling off cliffs.

#5: Error Prevention

New pipeline

Review your pipeline YAML

Variables | Save and run

PublicWeb / azure-pipelines.yml * | Show assistant

```yaml
# ASP.NET Core
# Build and test ASP.NET Core projects targeting .NET Core.
# Add steps that run tests, create a NuGet package, deploy, and more:
# https://docs.microsoft.com/azure/devops/pipelines/languages/dotnet-core

trigger:
- master

pool:
  vmImage: ubuntu-latest
  blah: test

variables:
  buildConfiguration: 'Release'

steps:
  - script: dotnet build --configuration $(buildConfiguration)
displayName: 'dotnet build $(buildConfiguration)'

```

#2 • Set up CI with Azure Pipelines

PublicWeb

Rerun failed jobs | Run new

This run will be cleaned up after 1 month based on your project settings.

Summary

Triggered by david.paquette

View 49 changes

Repository and version	PublicWeb master 25a2598c
Time started and elapsed	Just now <1s
Related	1 work items 0 artifacts
Tests and coverage	Get started

Errors 1

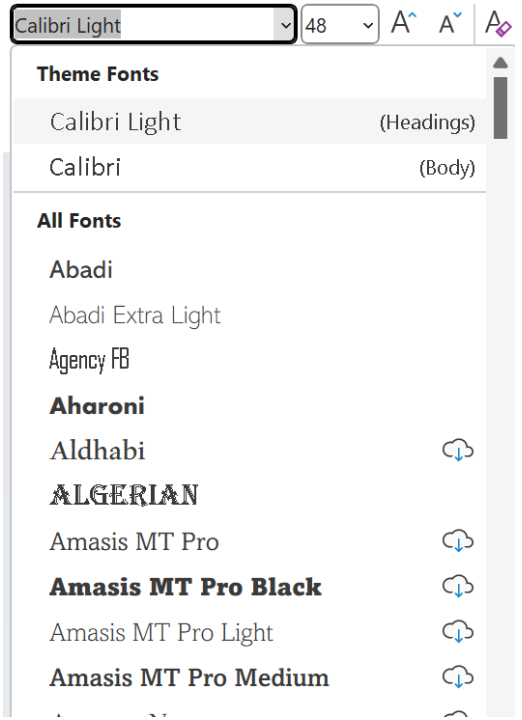
/azure-pipelines.yml (Line: 18, Col: 1): Unexpected value 'displayName'
2

Troubleshooting failed runs

#6: Recognition rather than recall

- Minimize user's memory load by making elements, actions, and options visible
- User should not have to remember information from one part of the interface to another
- Information required to use the design (e.g. field labels or menu items) should be visible or easily retrievable when needed

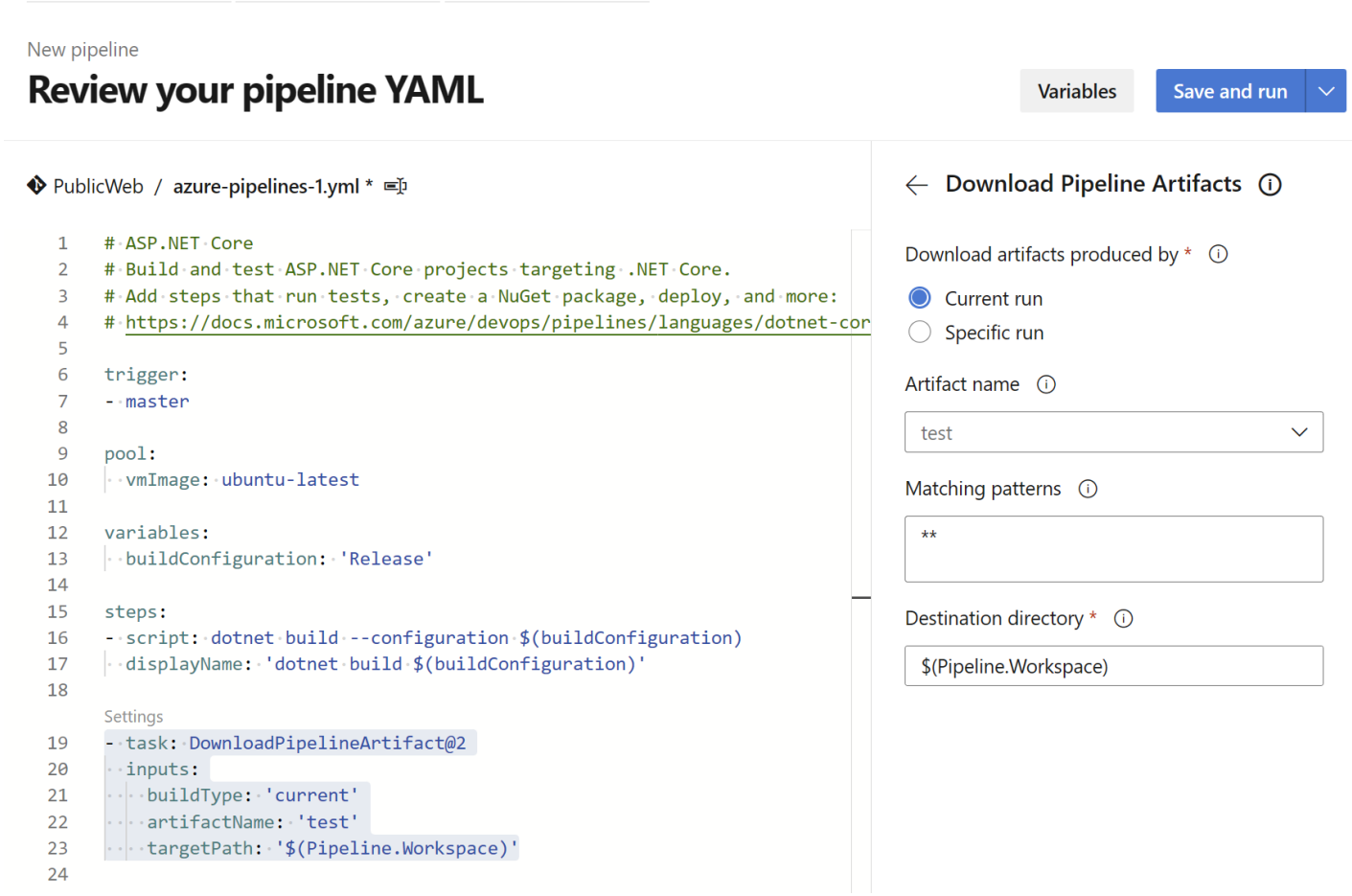
Computers are good at remember thing, people are not



https://www.nngroup.com/articles/recognition-and-recall/

#6: Recognition rather than recall

- Most recently used (MRU) list
- Auto Complete
- Offer help in-context



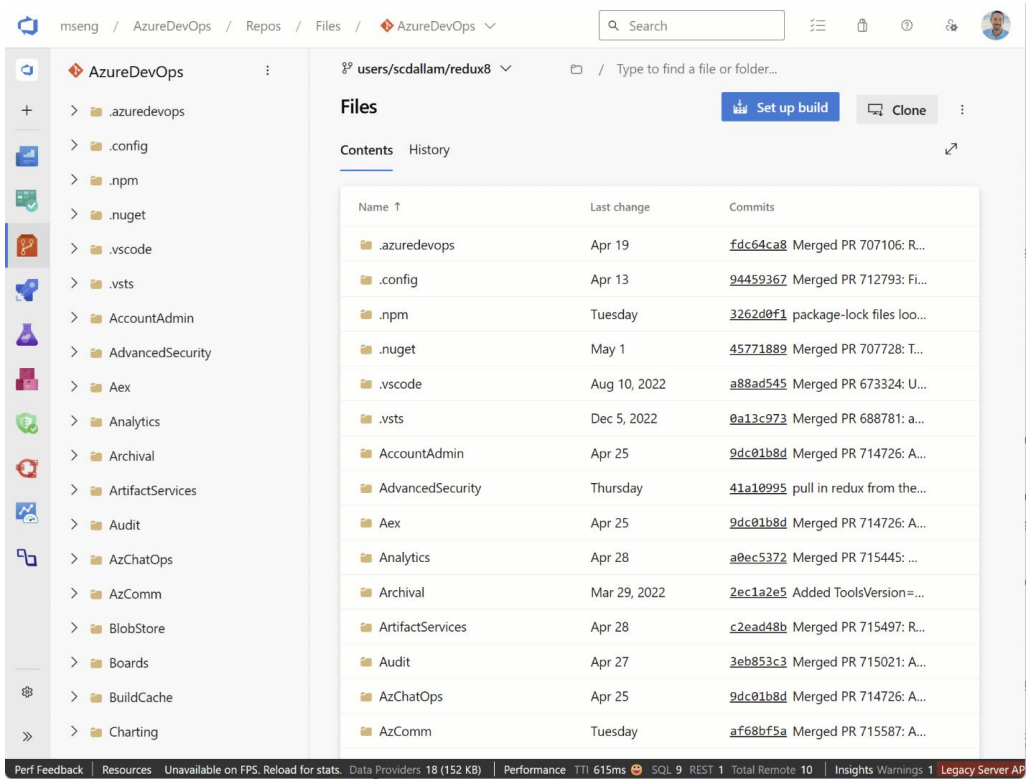
#7: Flexibility and efficiency of use

- **Shortcuts** — hidden from novice users — may speed up the interaction for the expert user so that the design can cater to both inexperienced and experienced users
- Allow users to **tailor frequent actions**

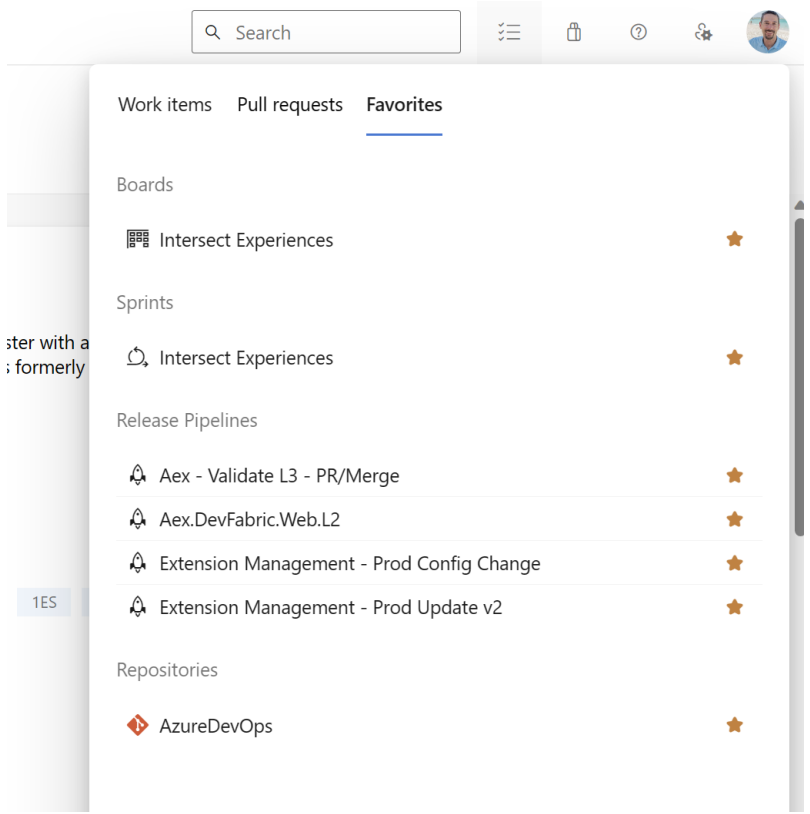
e.g. Regular routes are listed on maps, but locals with knowledge of the area can take shortcuts.

#7: Flexibility and efficiency of use

Press . on any repository page



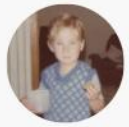
Favourites



#8: Aesthetic and minimalist design

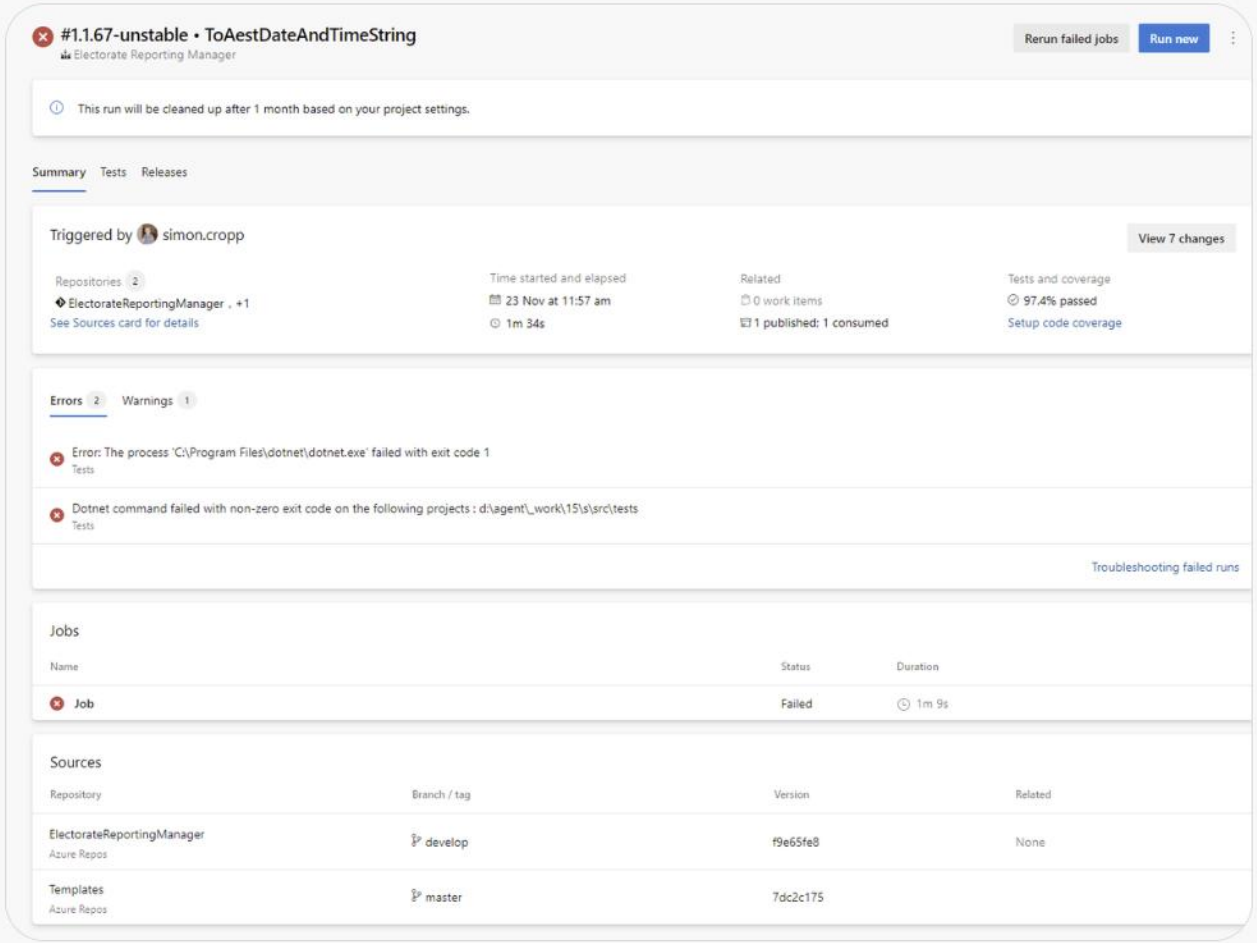
- Interfaces should not contain information that is irrelevant or rarely needed
- Every extra unit of information in an interface competes with the relevant units of information and diminishes their relative visibility

e.g. An ornate teapot may have excessive decorative elements, like an uncomfortable handle or hard-to-wash nozzle, that can interfere with usability.



Simon Cropp @SimonCropp · Dec 8, 2022

all this information for a broken build. and all i really want is to see the broken tests, which is a tiny link for a different tab

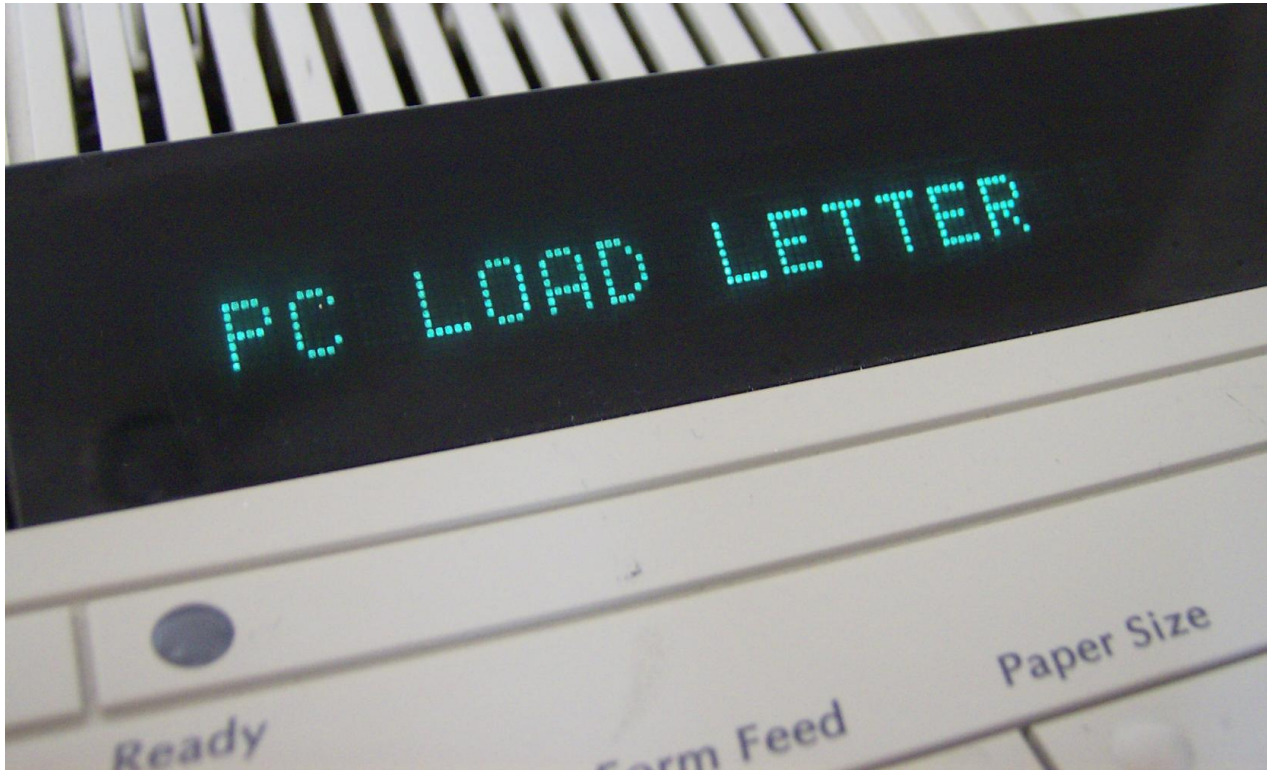


1

#8: Aesthetic and minimalist design

#9: Help users recognize, diagnose, and recover from errors

- Error messages should be expressed in plain language (no error codes), precisely indicate the problem, and constructively suggest a solution.



#9: Help users recognize, diagnose, and recover from errors

Errors 1

/.azuredevops/pipelines/builds/tfs_l2_a11y.selfhost.yml (Line: 3, Col: 17): Pipeline Resource source-pipeline Input Must be Valid.
$(BuildDefinitionName)_$(Build.SourceBranchName)_$(Date:yyyyMMdd)$(Rev:.r)

```yaml
resources:
  pipelines:
    - pipeline: source-pipeline # Name of the pipeline resource.
      source: AzureDevOps # The name of the pipeline referenced by this pipeline
      trigger:  # Run this pipeline when any run of AzureDevOps completes
        branches:
          exclude:
          - releases/*
          - master
          include:
          - users/dpaquette/ado-a11y-shift-left-spike

```

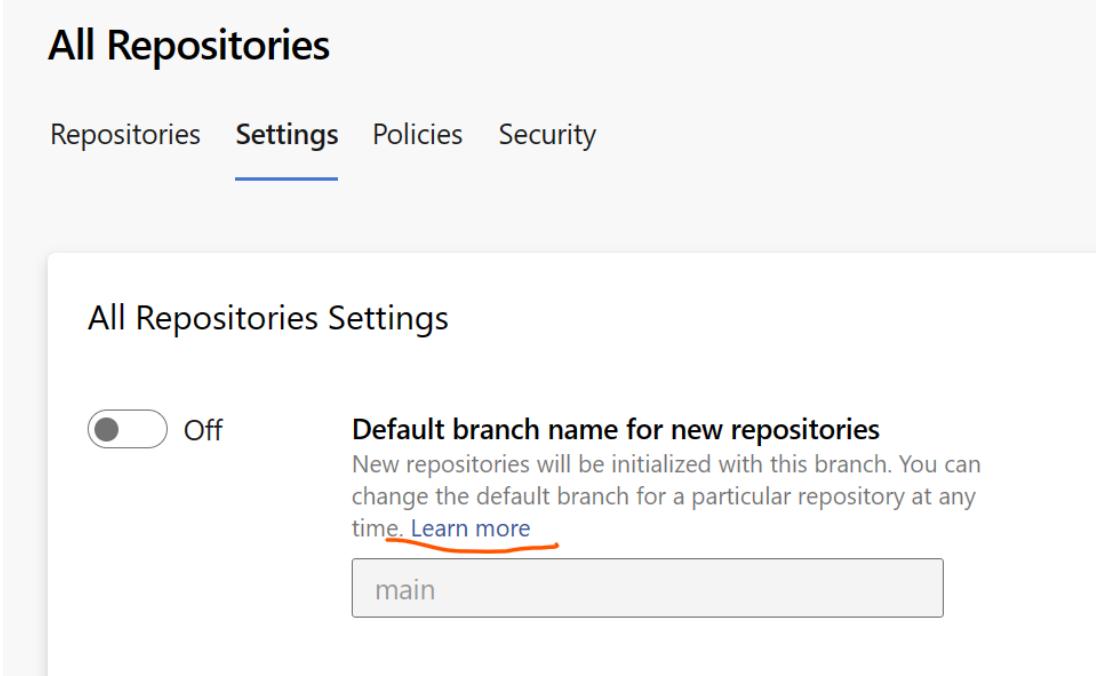
#10: Help and documentation

- It’s best if the system doesn’t need any additional explanation. However, it may be necessary to provide documentation to help users understand how to complete their tasks.

e.g. Information kiosks at airports are easily recognizable and solve customers’ problems in context and immediately.

#10: Help and documentation

- Ensure that the help documentation is easy to search
- Whenever possible, present the documentation in context right at the moment that the user requires it.
- List concrete steps to be carried out.



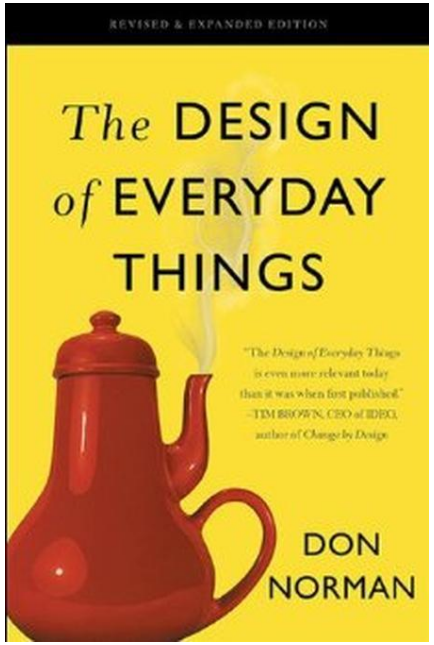
Key Points

- Speak the user's language
- Be consistent
- Keep it simple
- Talk to users if you can

It's not the user's fault!!!!!

References

https://www.nngroup.com/articles/ten-usability-heuristics/

https://www.nngroup.com/

The Design of Everyday Things – Donald Norman