

Development pipelines

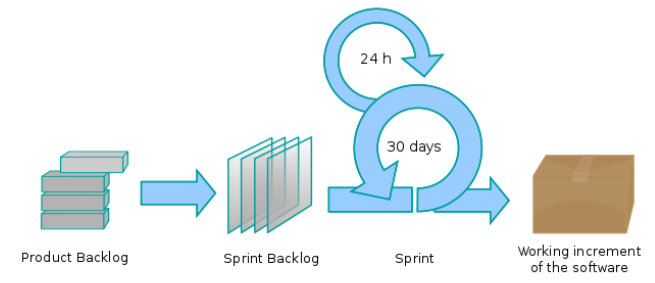
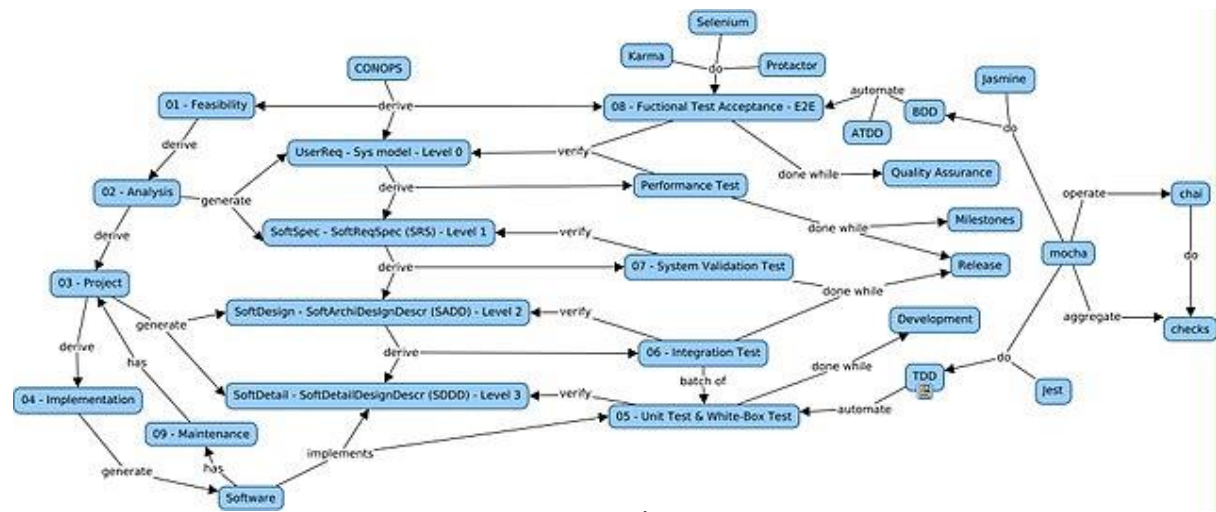
From code commit to shipping it



There cannot be a more important thing for an engineer, for a product team, than to work on the systems that drive our productivity.

So I would, any day of the week, trade off features for our own productivity.

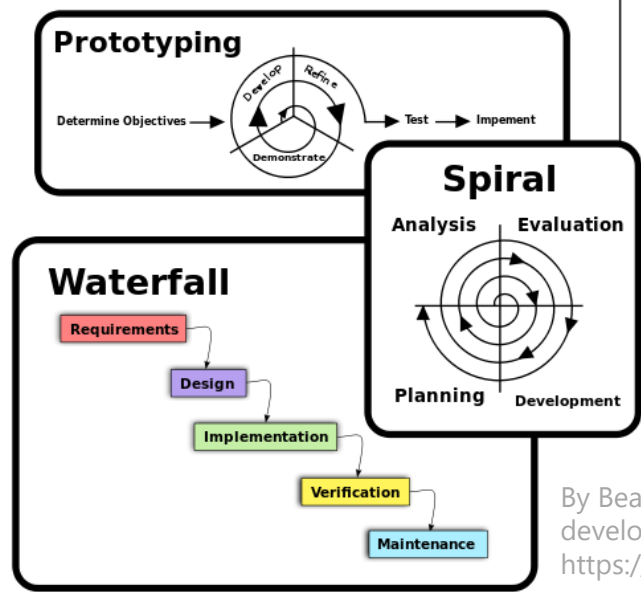
I want our best engineers to work on our engineering systems, so that we can later on come back and build all of the new concepts we want.



By Lakeworks - Own work, CC BY-SA 4.0, <https://commons.wikimedia.org/w/index.php?curid=3526338>

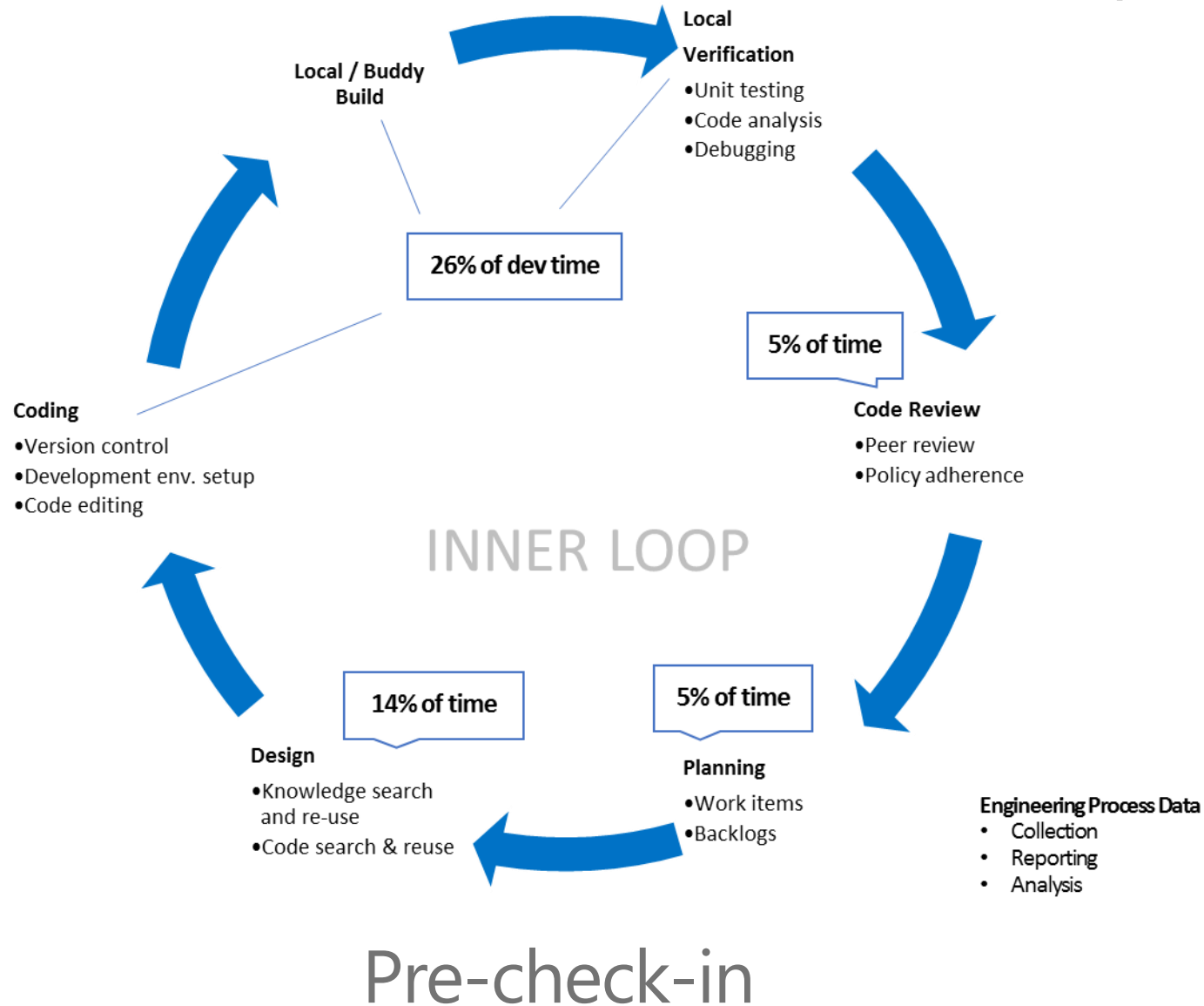
By Mark - Own work, CC BY-SA 4.0, <https://commons.wikimedia.org/w/index.php?curid=68365751>

Development Processes

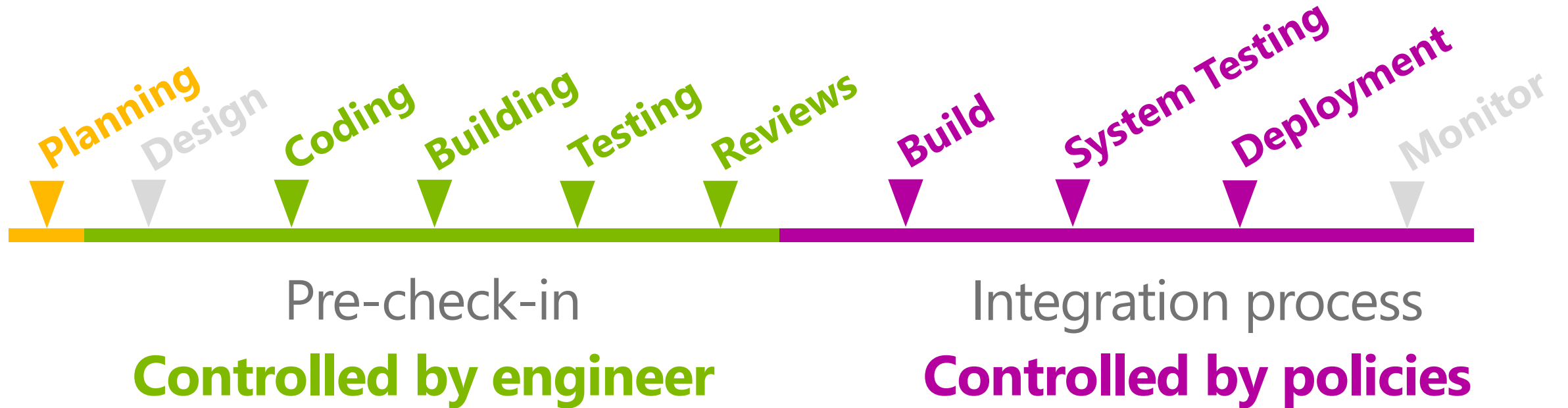


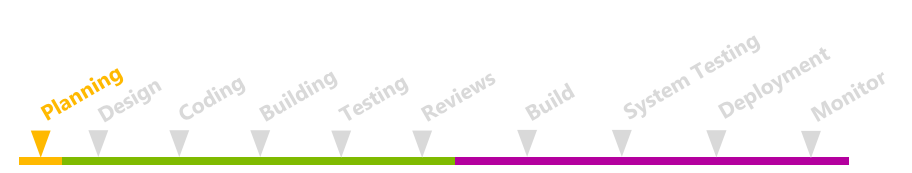
By Beao Old waterfall: Paul Smith - File:Waterfall model revised.svgFile:Rapid application software development.svgFile:Software Development Spiral.svg, Public Domain, <https://commons.wikimedia.org/w/index.php?curid=7836950>

Inner & Outer Development Loop



One Iteration





Phase 1
Planning

Objectives and Key Results: OKR

Methodology

- Set quarterly goals
- Measure metric progress
- Share with leaders & co-workers

Goals

- Common goal set
- Teams are aligned and linked
- Transparency to everyone

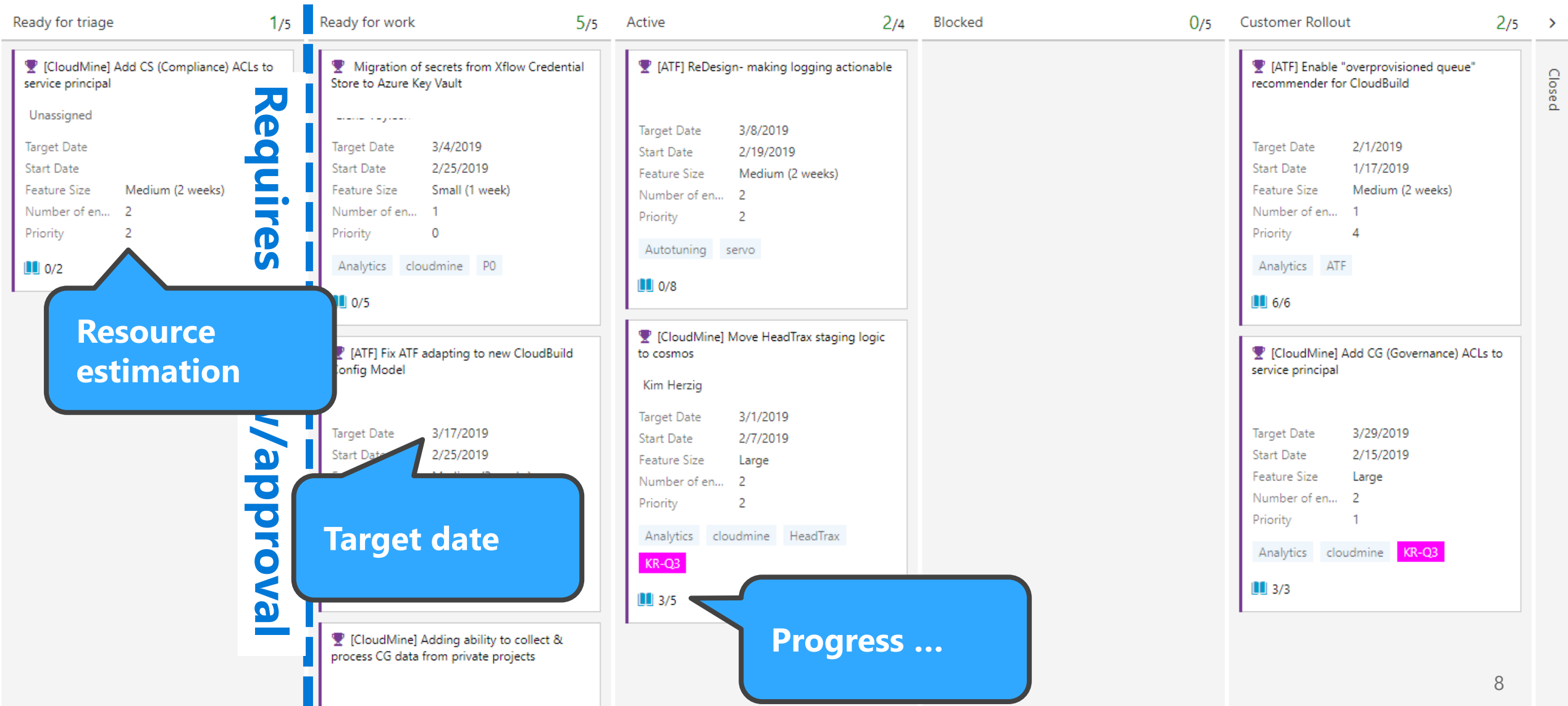
Objective:

80% of Microsoft integration builds performed using CloudBuild.

Key Results:

- Select top 10 teams not utilizing CloudBuild by March.
- Perform gap analysis for these teams by April.
- Implement 90% of required features in CloudBuild by July.
- Start migration process and transition 80% of builds to CloudBuild before December.

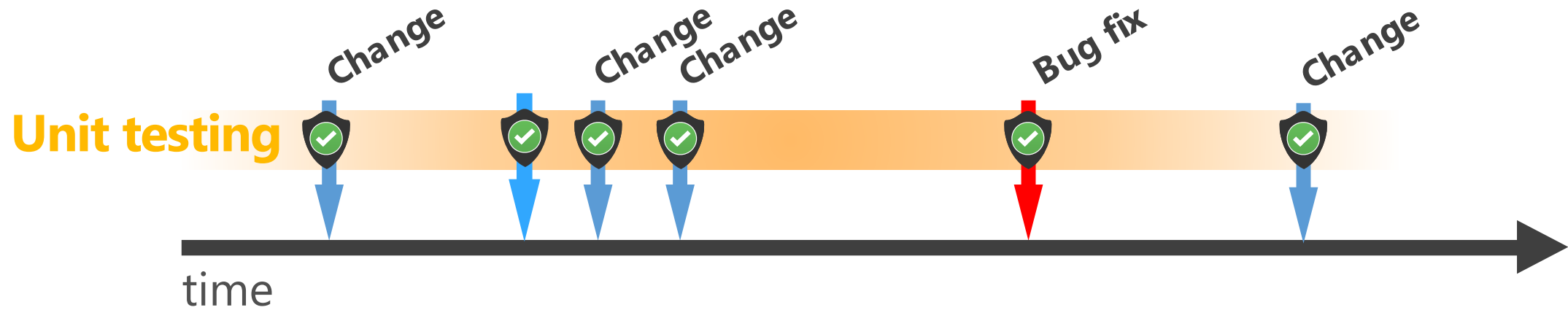
Kanban





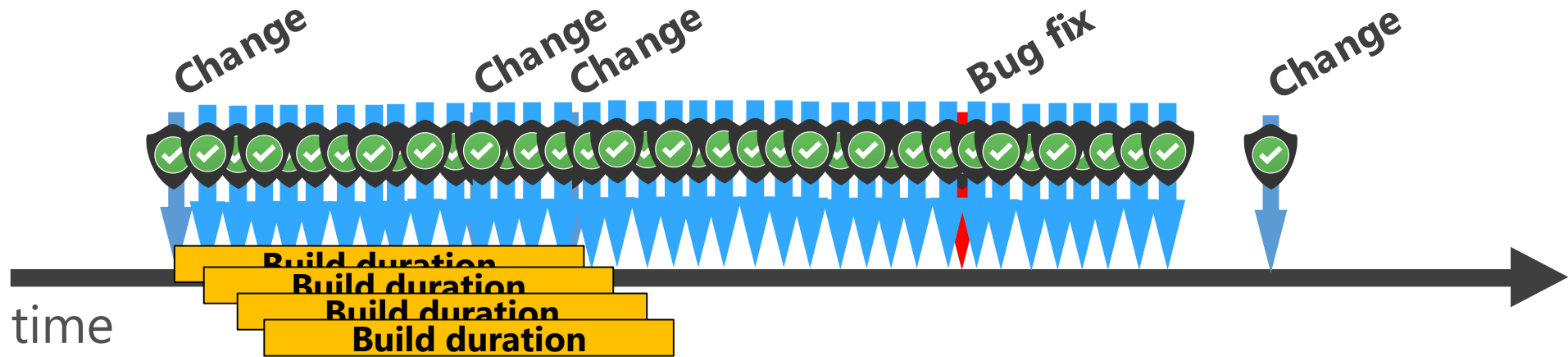
Phase 2
Coding

Simple Scenario



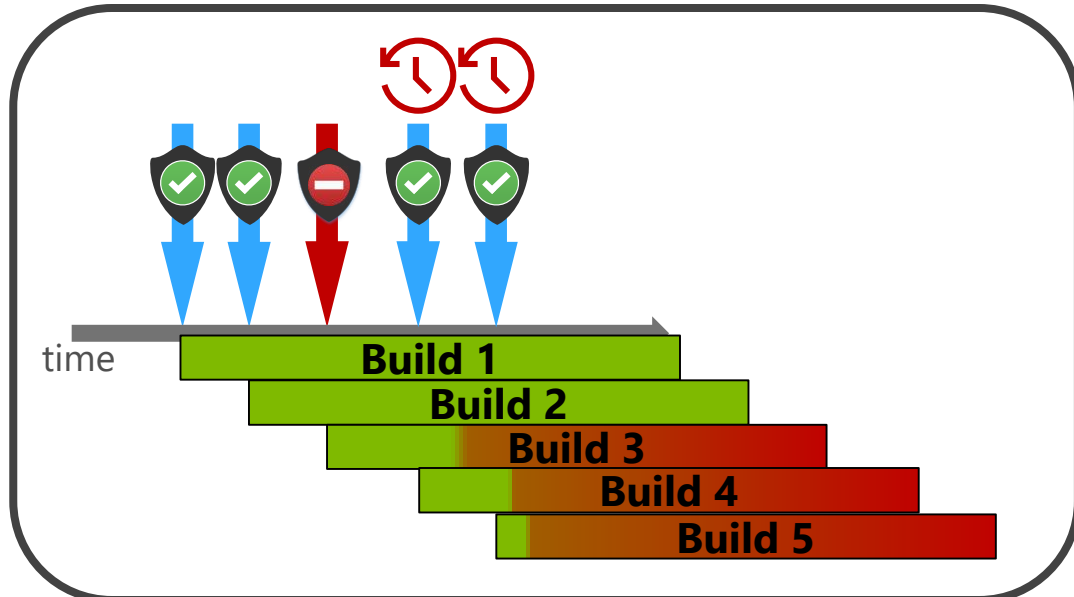
- Code changes are applied into a single repository.
- Code changes are immediately visible for everybody.
- Unit tests check for functional correctness at function/method level
- Developer run "their" tests.

But ...

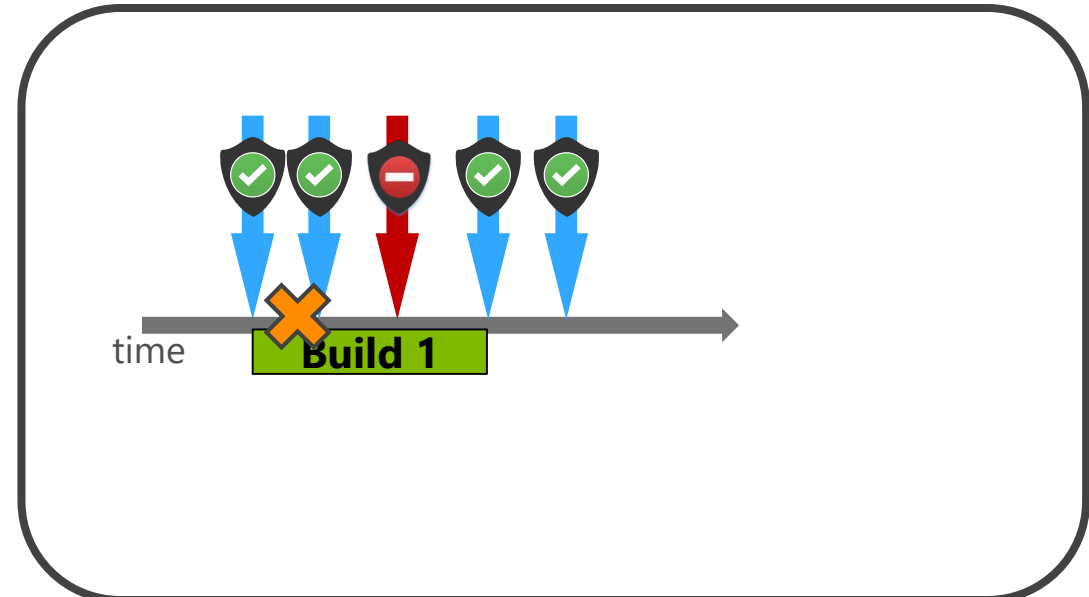


- Thousands of engineers.
- Millions of lines of code and millions of changes.
- Different organizational groups distributed around the world.
- Shared code: Windows Desktop, Server, Phone, Azure, Xbox, ...

Repo strategies

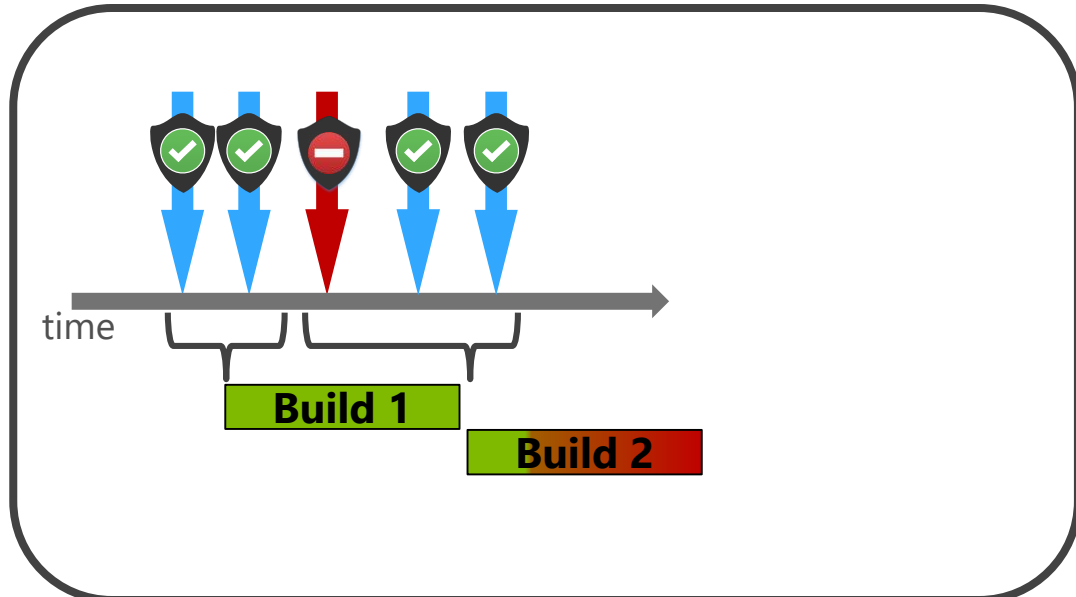


Overlay: Works well on componentized products with clear interfaces.

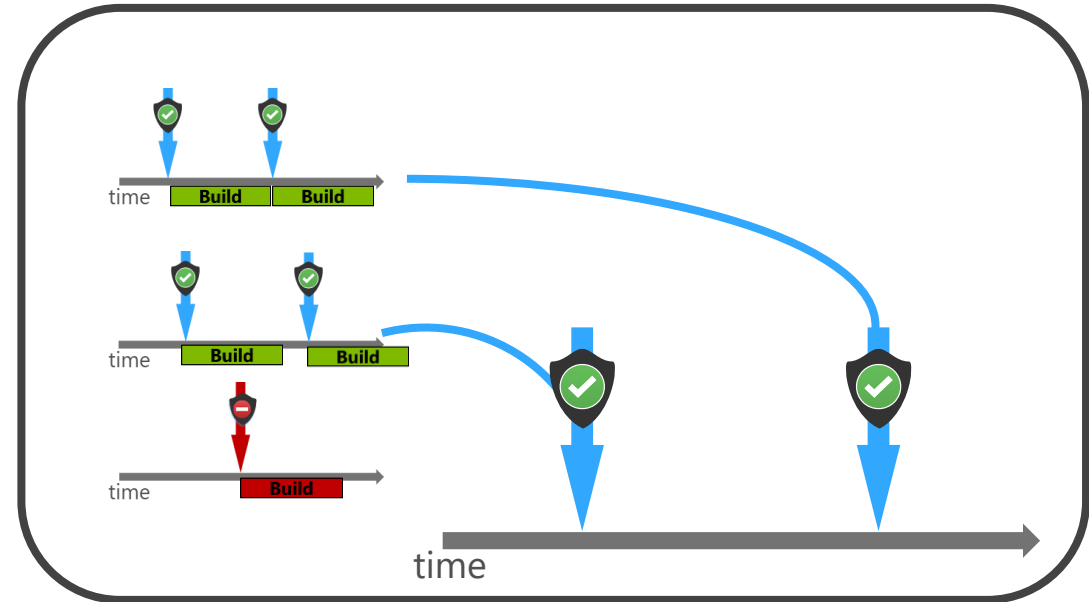


Locking: Creates long build queues and bottlenecks.

Repo strategies



Rolling: Fast, but when failing hard to find issue.

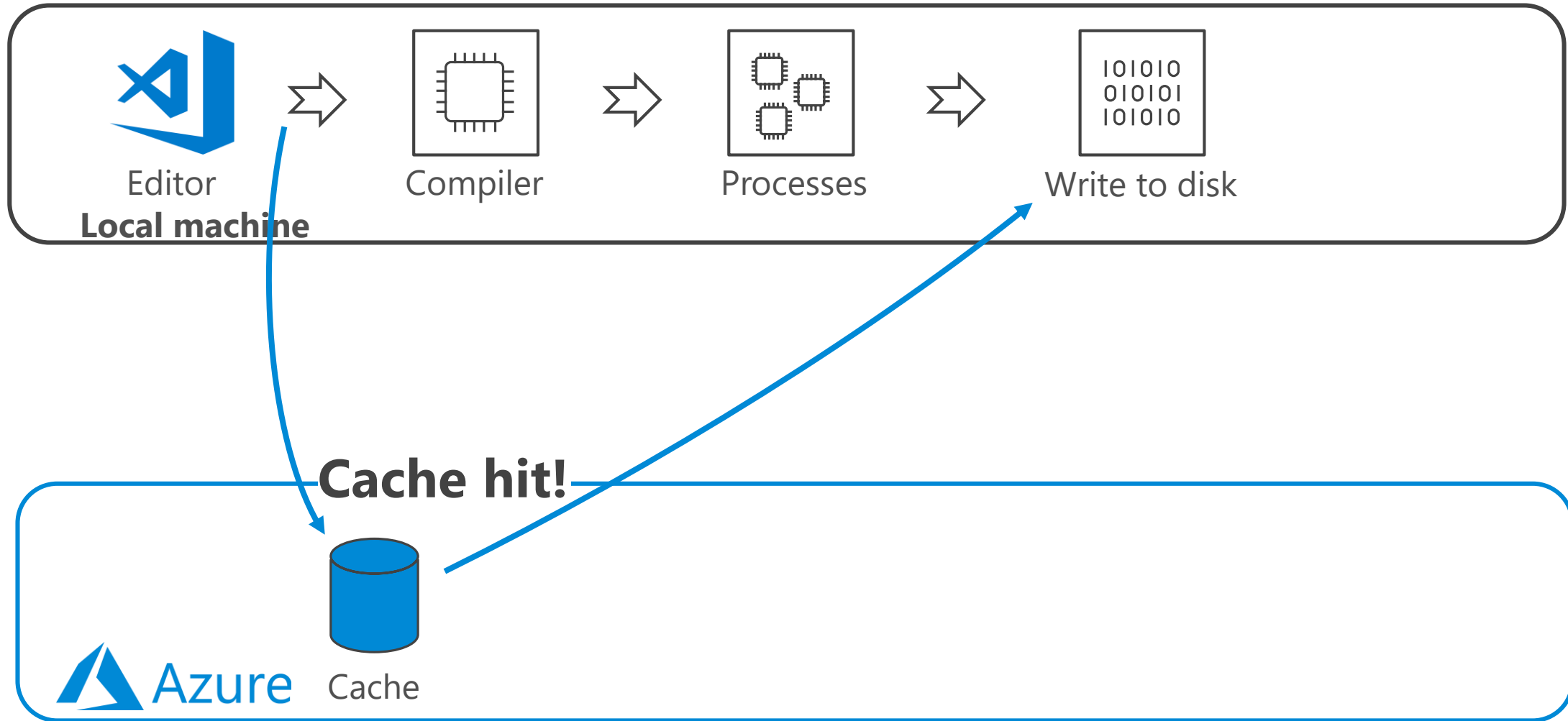


Branching: Can get very complex and slow.

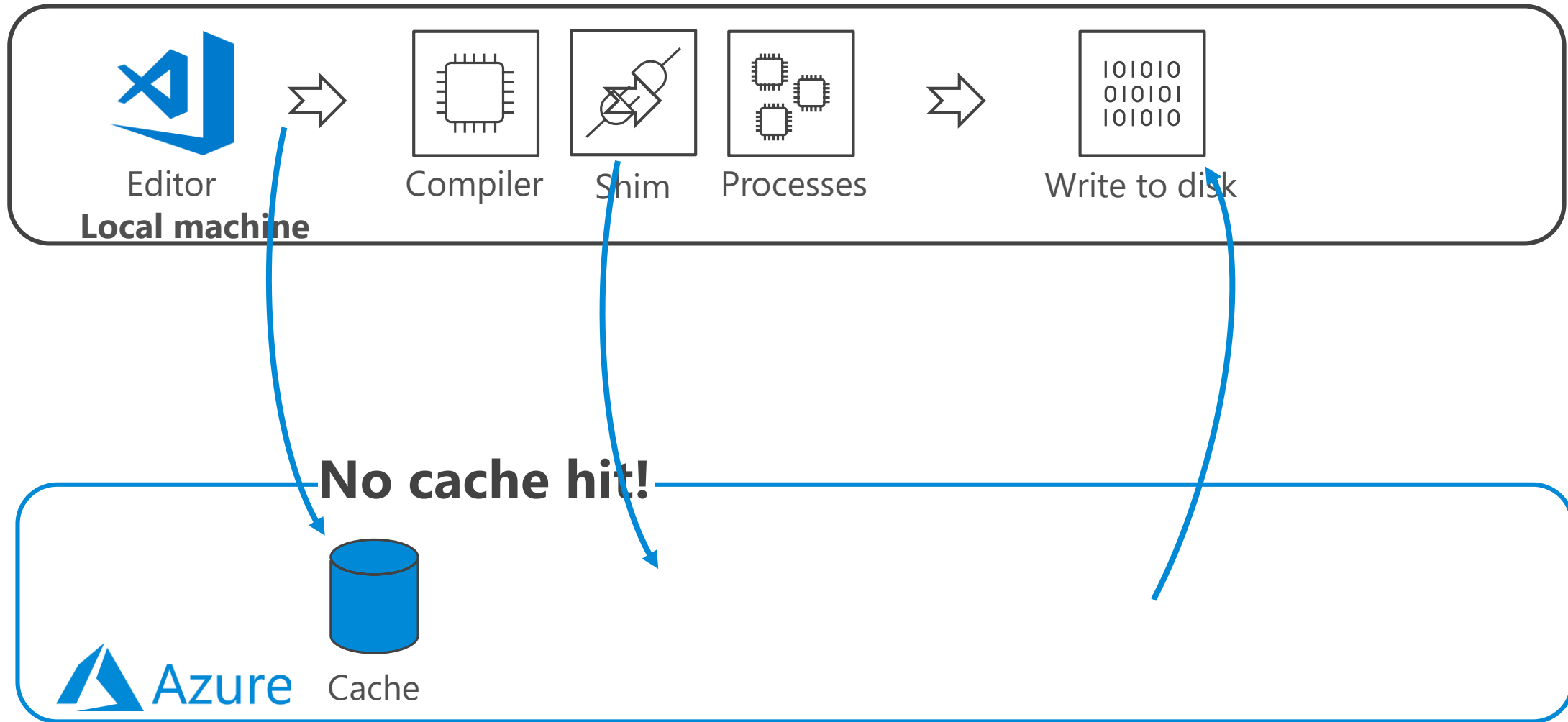


Phase 3
(Local?) Building

Local (?) builds



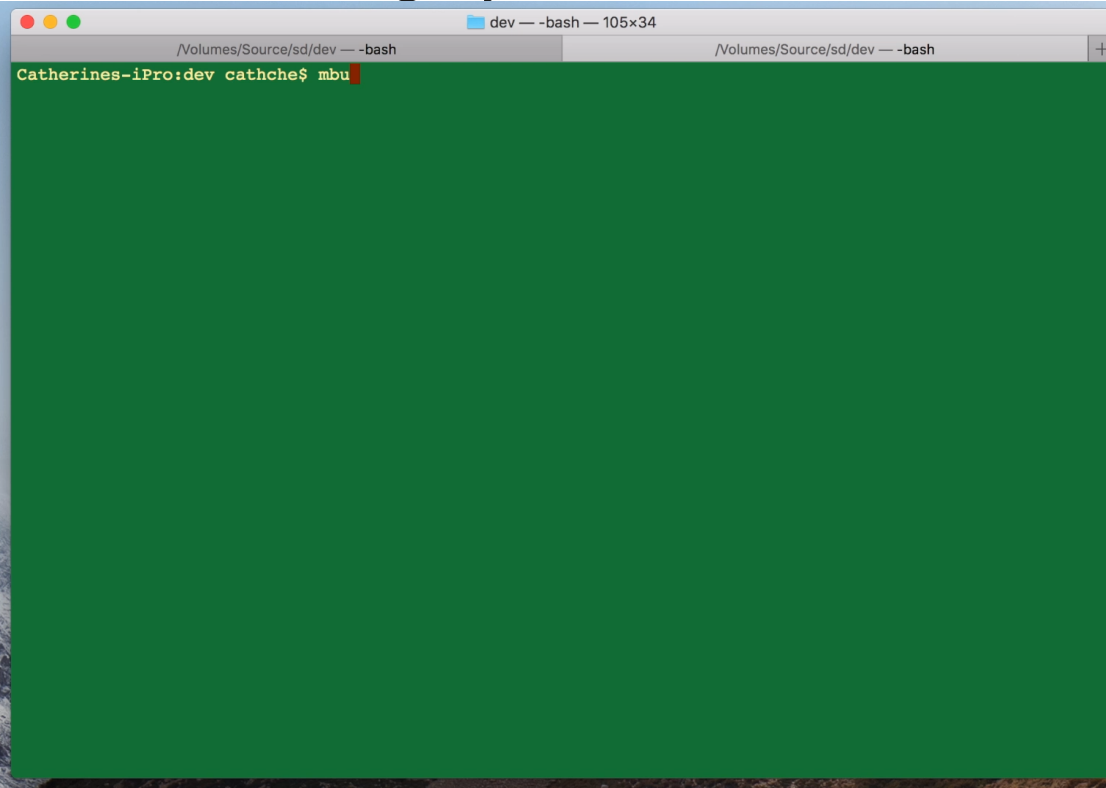
Local (?) builds



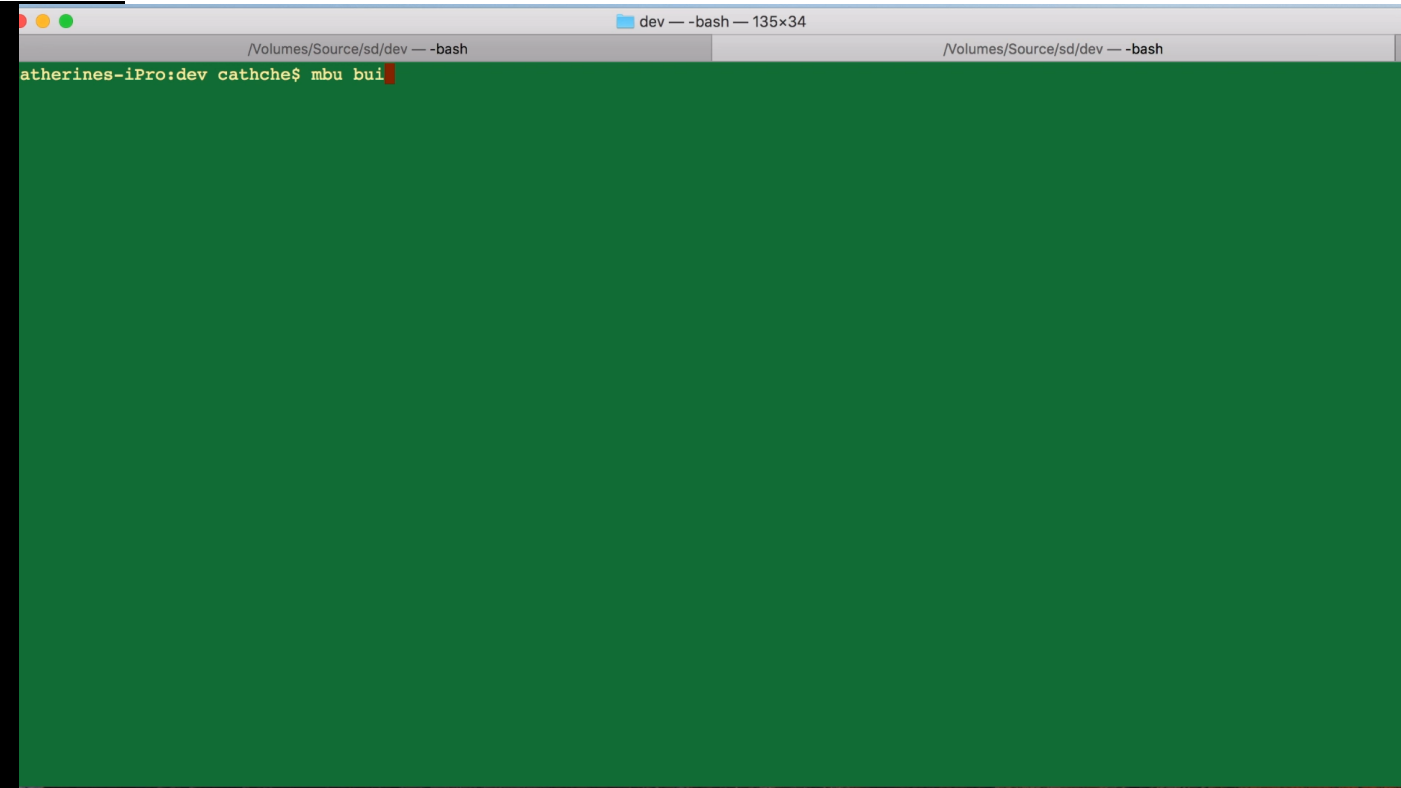
Cloud Build demo – Project-level build

legacy build 4x

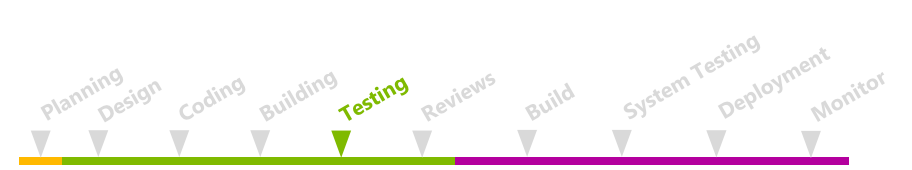
cloud build 4x



A terminal window titled "dev - bash - 105x34" showing a shell prompt "Catherines-iPro:dev cathche\$ mbu" with a red cursor. The terminal content is obscured by a solid green rectangle.



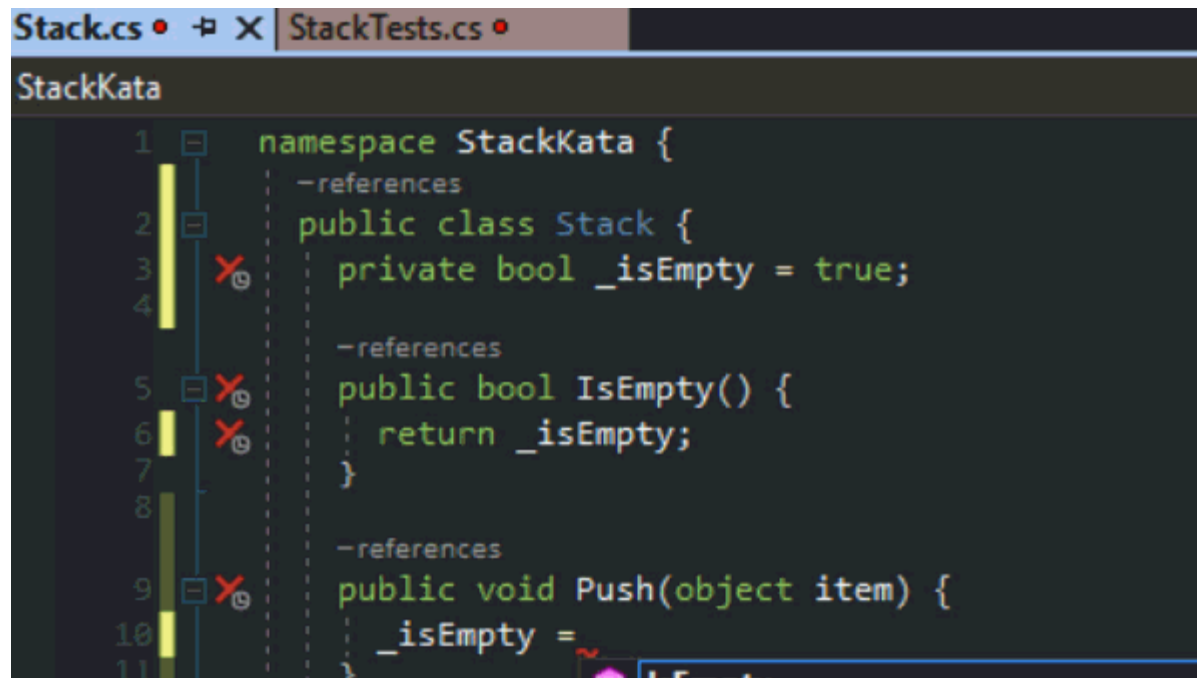
A terminal window titled "dev - bash - 135x34" showing a shell prompt "atherines-iPro:dev cathche\$ mbu bui" with a red cursor. The terminal content is obscured by a solid green rectangle.



Phase 4
Unit-Testing

Local Unit Testing

As you write code Live Unit Testing automatically runs any impacted unit tests in the background and presents the results and code coverage in real time.



```
Stack.cs • StackTests.cs •
StackKata
1 namespace StackKata {
2     -references
3     public class Stack {
4         private bool _isEmpty = true;
5         -references
6         public bool IsEmpty() {
7             return _isEmpty;
8         }
9         -references
10        public void Push(object item) {
11            _isEmpty =
12            IsEmpty;
```



Phase 5
Code Reviews

Branch Policies

Review required

Work item link?

Enforce squash commit

Builds need to pass ...

Protect this branch

- Setting a Required policy will enforce the use of pull requests when updating the branch
- Setting a Required policy will prevent branch deletion
- Manage permissions for this branch on the [Security page](#)

Require a minimum number of reviewers

Require approval from a specified number of reviewers on pull requests.

Minimum number of reviewers

- Allow users to approve their own changes.
- Allow completion even if some reviewers vote "Waiting" or "Reject".
- Reset code reviewer votes when there are new changes.

Check for linked work items

Encourage traceability by checking for linked work items on pull requests.

Policy requirement

- Required
Block pull requests from being completed unless they have at least one linked work item.
- Optional
Warn if there are no linked work items, but allow pull requests to be completed.

Check for comment resolution

Check to see that all comments have been resolved on pull requests.

Policy requirement

- Required
Block pull requests from being completed while any comments are active.
- Optional
Warn if any comments are active, but allow pull requests to be completed.

Enforce a merge strategy


Require a specific type of merge when pull requests are completed.

- No-fast-forward merge
The source branch and all its commits will appear as a second parent to the merge commit.
- Squash merge
Condense all the changes in a pull request into one commit with one parent. [Learn more](#)

Build validation

Validate code by pre-merging and building pull request changes

+ Add build policy

Build pipeline	Requirement	Path filter	Expiration	Trigger	
 CloudMine Cosmos PR	Required	No filter	Expires after 12 hours	Automatic	<input checked="" type="checkbox"/> Enabled
 CloudMine-DocFx-PR	Required	docfx_project/*	Expires after 12 hours	Automatic	<input checked="" type="checkbox"/> Enabled

Pull Requests

437299 **ACTIVE** Fixing project files and adding

Kim Herzig dev/kimh/repo_cleanup into master

Overview Files Updates Commits Additional Validations Co

Description

- Fixing broken projects from last PR
- Adding `dirs.proj` that replaces (replaced) two build step phases in this is fixing it.
- To run use `msbuild /restore /p:FakeDeployment=true /t:Build,De`

Once this is in master I will change the build definitions to use this dirs

: The binaries for VSTSAalytics do not get automatically the binaries into one, but is there a reason to prevent your binaries fro

Show everything

Add a comment...

5 Kim Herzig pushed 1 commit creating update 5

09eadb4c fixing build task for vscode 3 hours ago

4 Kim Herzig pushed 1 commit creating update 4

c386c8df fixing assemblies.xflowproj 3 hours ago

Kim Herzig 3 hours ago

Policies

Required

- ✗ 0 of 1 reviewers approved
- ✗ Required reviewers have not approved
- ✓ All comments resolved
- ✓ Build succeeded

Optional

- ✗ No work items linked

Approve Set auto-complete

Policies

- Required
 - ✗ 0 of 1 reviewers approved
 - ✗ Required reviewers have not approved
 - ✓ All comments resolved
 - ✓ Build succeeded
- Optional
 - ✗ No work items linked

Work Items

No related work items

Reviewers

Required

- Analytics

Optional

- HS
- KL

Live Share

```
File Edit Selection View Go Debug Terminal Help • addExample.js - livesharedemo - Visual Studio Code
```

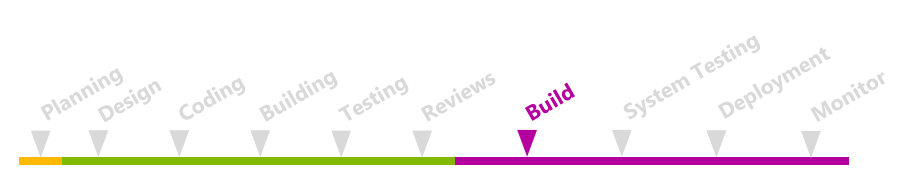
```
JS addExample.js •  
7 async function add(number) {  
8   if (number < 0) {  
9     throw new Error('The number must be positive');  
10  }  
11  var a = resolveAfter50ms(60);  
12  var b = resolveAfter50ms(30);  
13  return number + await a + await b;  
14 }  
15  
16 add(10).then(result => {  
17   // res[  
18 });
```

Ln 17, Col 5 Spaces: 4 UTF-8 CRLF JavaScript ✓ 0.90ms

```
File Edit Selection View Go Debug • addExample.js - Visual Studio Live Share (Workspace) - Visual S...
```

```
JS addExample.js •  
7 async function add(number) {  
8   if (number < 0) {  
9     throw new Error('The number must be positive');  
10  }  
11  var a = resolveAfter50ms(60);  
12  var b = resolveAfter50ms(30);  
13  return number + await a + await b;  
14 }  
15  
16 add(Simon McEnlly result => {  
17   // res  
18 });
```

Ln 1, Col 1 Spaces: 4 UTF-8 CRLF JavaScript ✓ 0.90ms



Phase 6

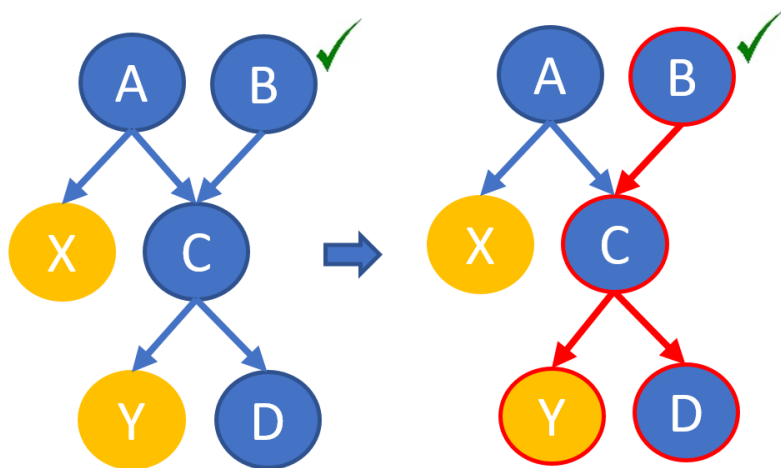
Integration Builds

CloudBuild Build-Graph

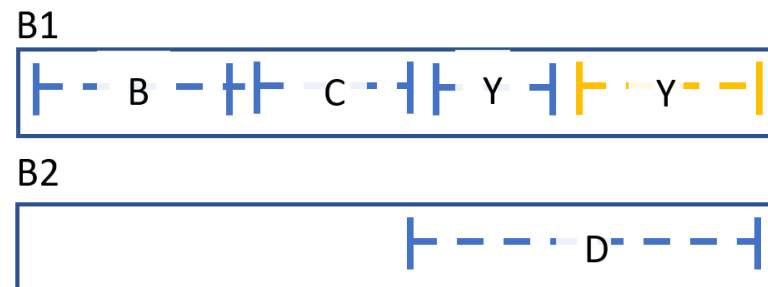
40,000 feet overview

- Prior to build: build dependency graph (DAG).
- Models dependencies between projects.
- Hashing compiler inputs ...
- ... rebuilds only targets for which hash lookup failed
- Distributed *build tasks* across machines and CPUs

1. Build Graph Analysis

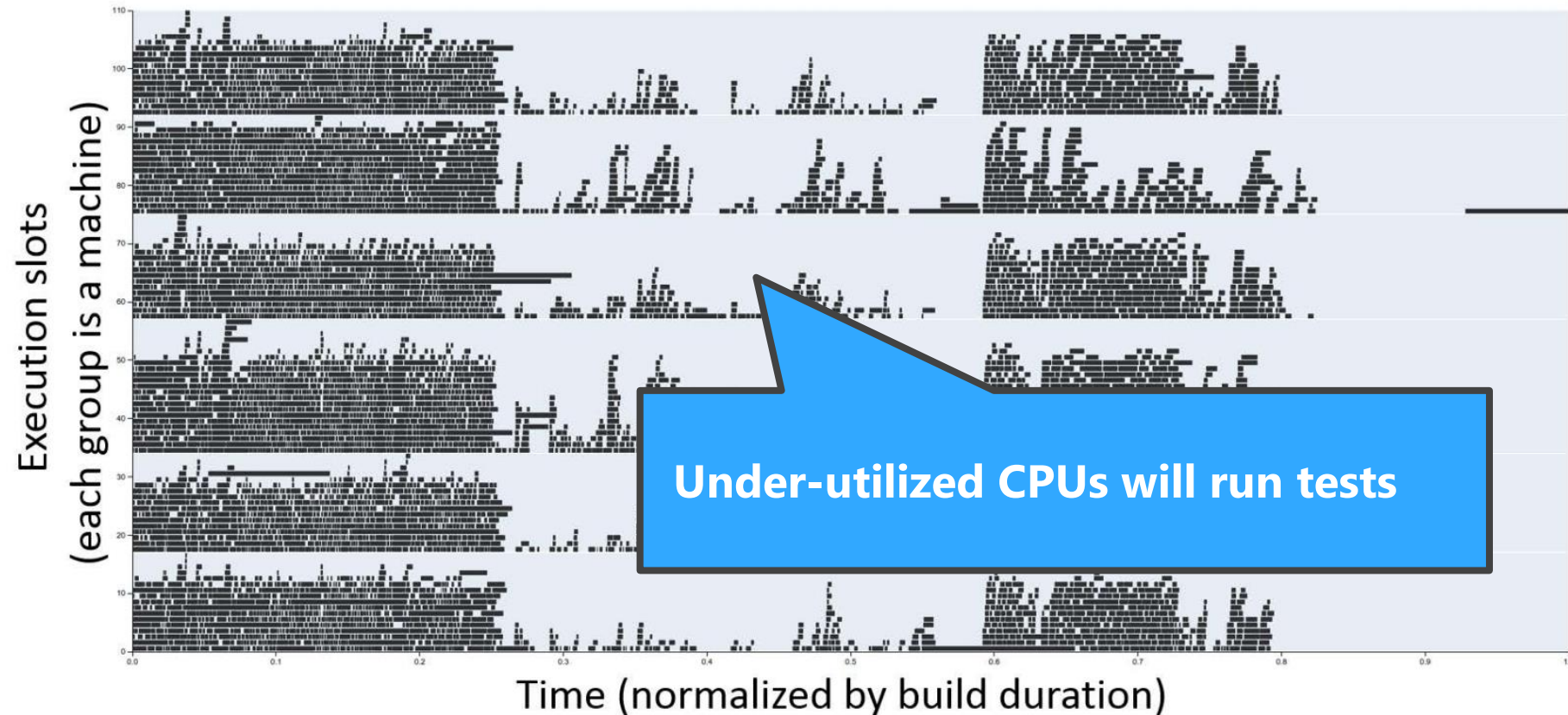


2. Build/Test Execution



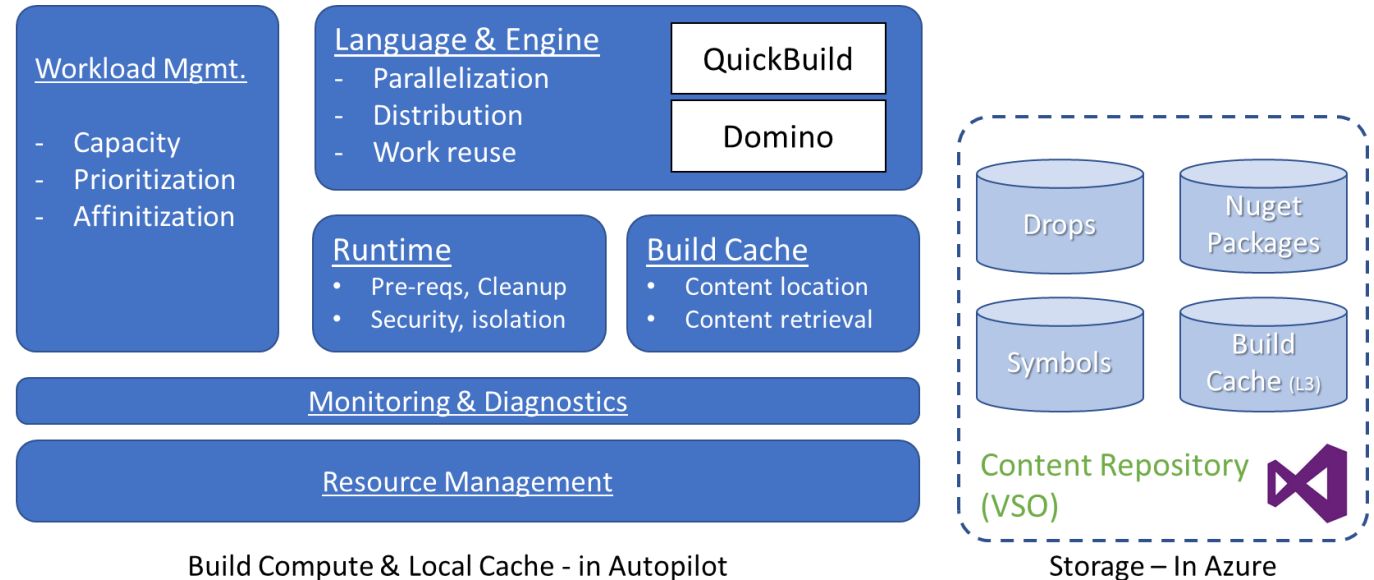
Unit-testing

- Tests run in parallel of build tasks (usually using under-utilized CPUs)
- Only run tests for projects that will build.



Integration Builds: CloudBuild

- Build
- Unit test
 - Code Coverage
 - Automatic retry
 - Flaky test management
- Static analyses
 - Code smell
 - Security vulnerabilities
 - Bad code behavior, e.g. leap year issue
 - Audit logs
- Code Signing



Integration Builds: CloudBuild

~70k

Branches

/ month

>1m

Builds

/ month

~70%

Cache hit

/ build

~2.5 bn

DLLs

/ month

~55%

Tests cached

>81m

Test suites

/ month

>9bn

Test cases

/ month

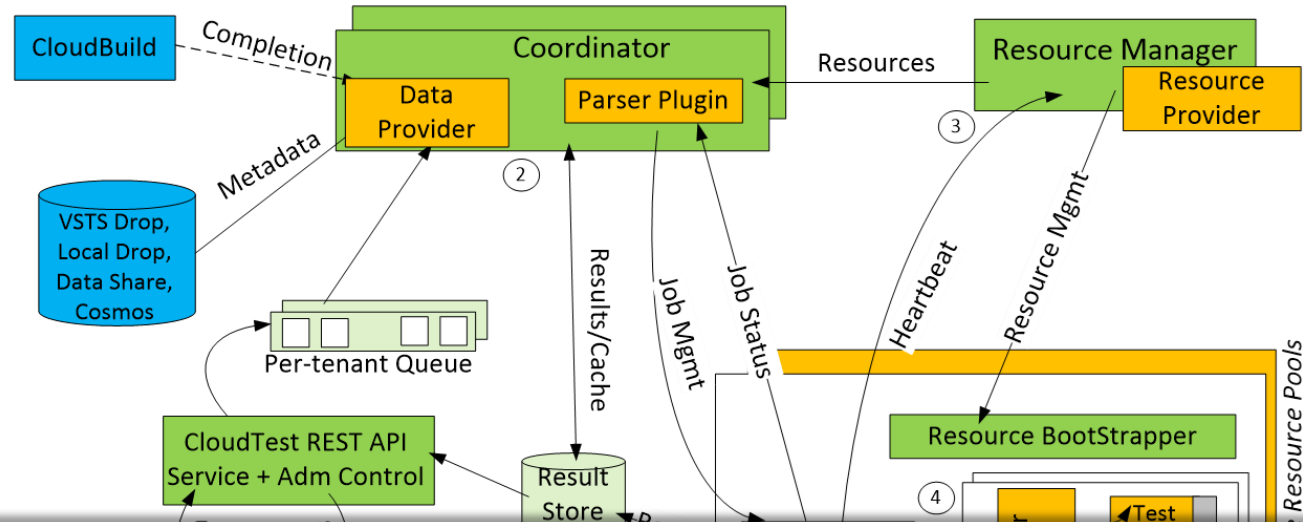
<0.05%

Failure rate



Phase 7
System Testing

Integration Testing: CloudTest 40,000 feet overview



8k

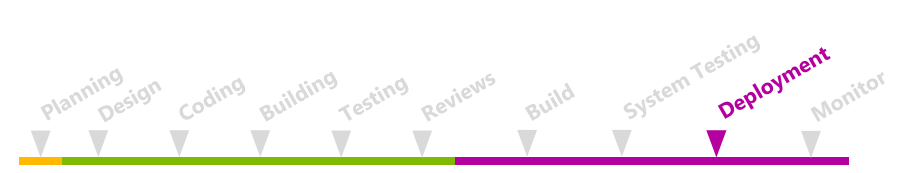
Users
/ month

6k

Definitions
/ month

1bn

Test case
executions
/ month



Phase 8
Deployment

Azure Pipelines

Cloud-hosted pipelines for Linux, Windows and macOS, with unlimited minutes for open source



Any language, any platform, any cloud

Build, test, and deploy Node.js, Python, Java, PHP, Ruby, C/C++, .NET, Android, and iOS apps. Run in parallel on Linux, macOS, and Windows. Deploy to Azure, AWS, GCP or on-premises



Flexible Workflows & Extensible

Explore and implement community-built build, test, and deployment tasks, along with hundreds of extensions from Slack to SonarCloud. Support for YAML, test integration, release gates, reporting, and more.



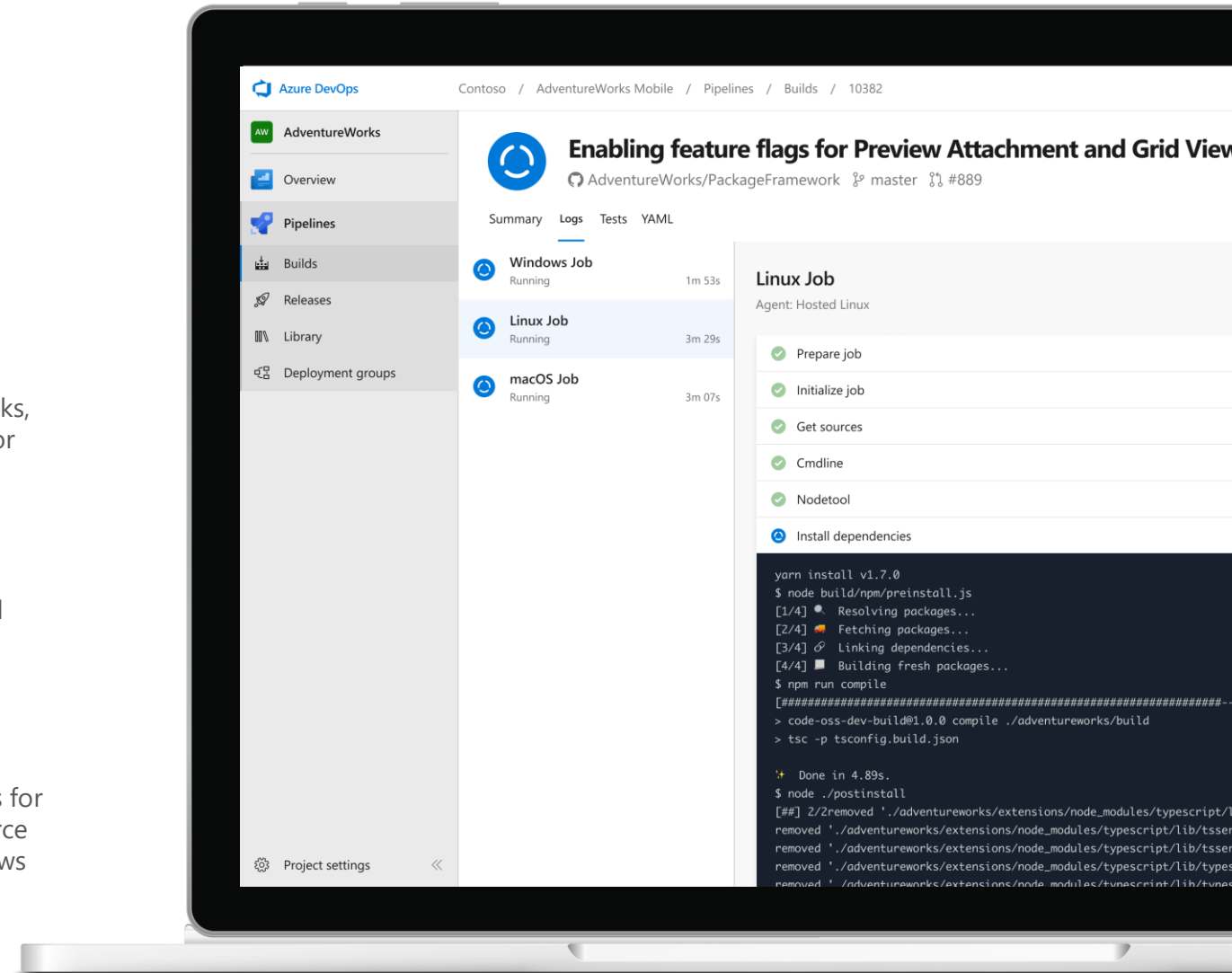
Containers and Kubernetes

Easily build and push images to container registries like Docker Hub and Azure Container Registry. Deploy containers to individual hosts or Kubernetes.



Best-in-class for open source

Ensure fast continuous integration/continuous delivery (CI/CD) pipelines for every open source project. Get unlimited build minutes for all open source projects with up to 10 free parallel jobs across Linux, macOS and Windows




Deployment Stages / Rings

Artifact → Validation → Dogfood → Canary (Ring 0) → ... → "Gold"-Ring


Release

Continuous deployment

for  Microsoft.VisualStudio...
2/26/2019, 3:13 PM

Artifacts



\$(Build.Definition... 
CloudBuild Deploy Service...
master

Stages

CI

✔ Succeeded

on 2/26/2019, 4:44 PM

CBTest

⚠ Partially succeeded

1 error
on 2/26/2019, 7:57 PM

Approve update of re

○ Not deployed

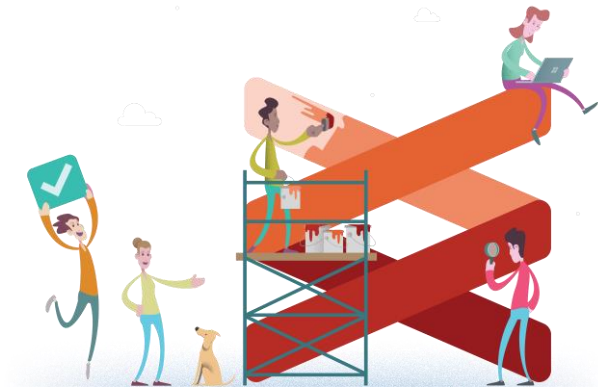
Validate ConfigCop be

✔ Succeeded

on 2/26/2019, 3:19 PM

Manual investigation

Sign-off



Shipped!
Let's do it again ... and again ...

Example Unit Tests:

Test Selection & Code Coverage

Example

Test selection based on code coverage

CODE
COVERAGE

*Slows down test
execution by 30%*

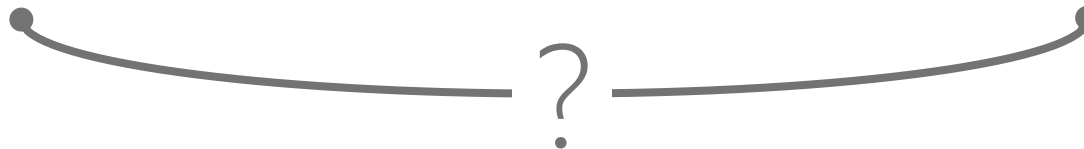


Your tool



RUNTIME
REDUCTION

Saving 50% of tests!



Let's do some basic math

Assumption: you achieve 50% test reduction

$$\begin{aligned} \text{Cost}_{\text{Test}} &= |\text{TestCases}| * \overline{\text{Runtime}_{\text{Testcase}}} * \text{Cost}_{\text{Machine}} \\ &= 10,000 * 0.12 \text{ sec} * 5.14e^{-4} \text{ \$/sec} = \mathbf{\$0.62 \text{ per build}} \end{aligned}$$

This excludes human effort!

$$\text{You saved: } \$0.62 * 2500 * 0.5 = \mathbf{\$722 \text{ per day}}$$

Product X:

- 10,000 unit test cases
- 0.12 sec per test case
- ~2500 builds per day
- Azure machine: \$1.85/hr

But ...

You used code coverage, right?

$$\begin{aligned} Cost_{CodeCoverage} &= (Cost_{Testing} * \overline{Runtime_{Overhead}}) + (CC_{Size} * Cost_{Disk}) \\ &\quad + (CC_{Size} * Speed_{I/O} * Cost_{Machine}) \\ &= (\$0.62 * 0.3) + (3.3GB * 0.35 \$/_{GB}) \\ &\quad + (2^{sec}/_{GB} * 3.3GB) * 5.14e^{-4} \$/_{sec} = \mathbf{\$1.34 \text{ per build}} \end{aligned}$$

You spend: $\$1.34 * 2500 = \mathbf{\$3,350 \text{ per day}}$

Facts:

- Block coverage: 3.3 GB per build
- 0.35 \$/GB SDD cost
- SSD speed 500MB/sec
- Azure machine: \$1.853/hr

It might still be worth the effort

But it's as good as you might think, and we have more pressing issues.

\$3,350



\$722

Slows down test execution by 30%

Saving 50% of tests!

And this excludes many aspects: analysis time, network cost, service maintenance, etc.

