



Atlassian Forge – Von der Idee zum Release

Daniel Kleissl

Disclaimer

- Kein:
 - React-Tutorial
 - In-Depth Forge-Tutorial
- Ein:
 - Forge-Tutorial für Einsteiger
 - Sprungbrett für die Entwicklung
 - Geteilter Erfahrungsschatz



Ressourcen

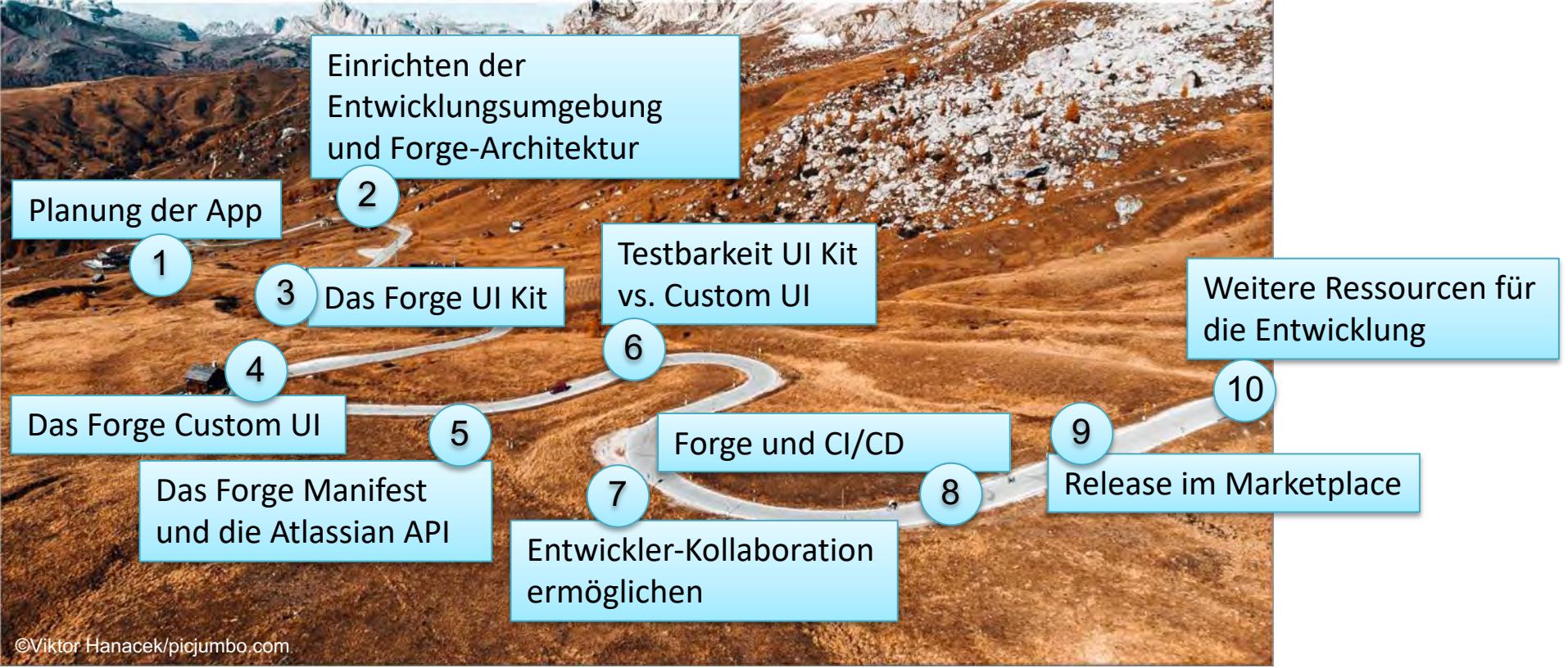
- Werden bereitgestellt:
 - Links
 - Code-Snippets



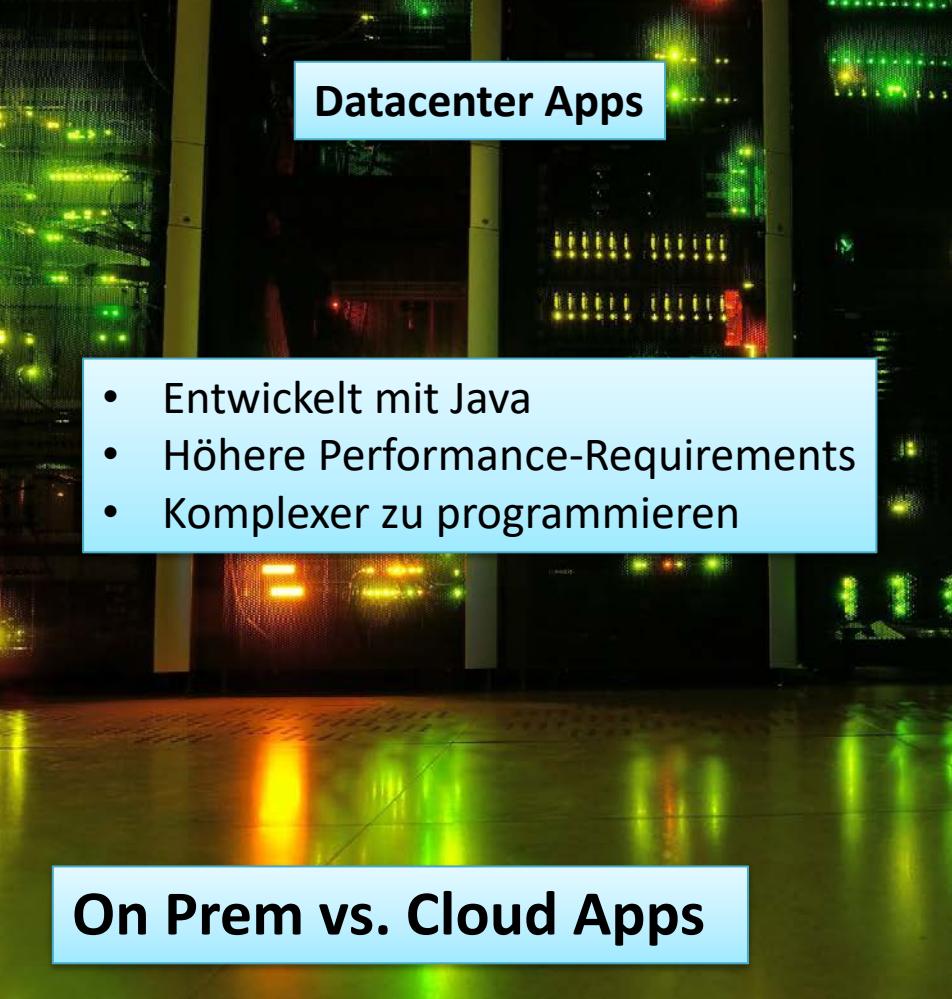
„Journey to Rome“ – Jira Risk Management

Berechnen Sie das Anforderungsrisiko und die Testintensität basierend auf 12 simplen Fragen und Antworten.

Weitere Informationen: www.razzfazz.io



©Viktor Hanacek/picjumbo.com



Datacenter Apps

- Entwickelt mit Java
- Höhere Performance-Requirements
- Komplexer zu programmieren



Cloud Apps

- Entwickelt mit JavaScript
- Über Cloud verteilt
- Einfacher zu programmieren

On Prem vs. Cloud Apps

©Viktor Hanacek/picjumbo.com

Serverless app
development *platform*

Beginner

Für:

- Confluence Cloud 
- Jira Cloud 
- Jira Service Management Cloud 

Everyone

FaaS – *Function as a
Service*

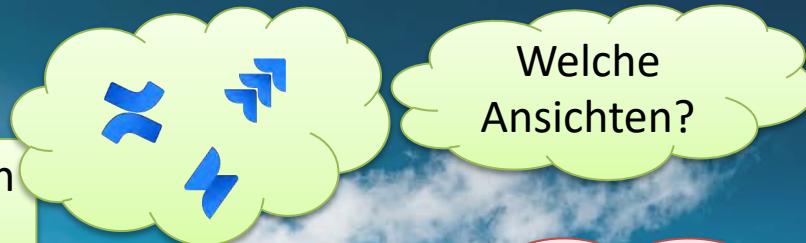
Advanced

Was ist Forge?

©Viktor Hanacek/picjumbo.com



Welches Atlassian
Produkt?

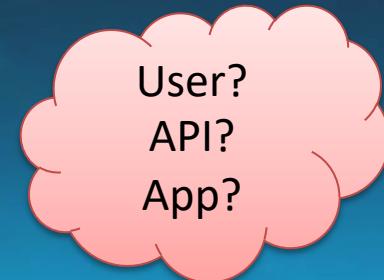


Welche
Ansichten?

App-
Funktionalität



Nur Anzeige?
Persistente
Datenänderung?
Intern vs. Extern?



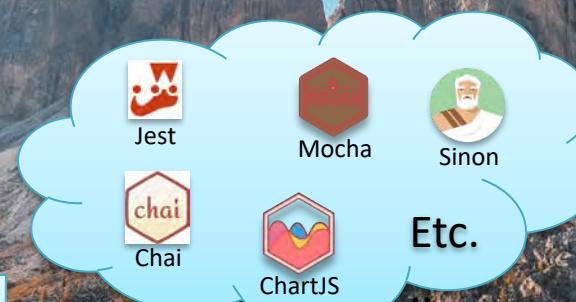
User?
API?
App?

Berechtigungen



Welcher Stack?
Welche Frameworks?

Plane den Aufstieg



test task

[Attach](#)[Create subtask](#)[Link issue](#)[Journey To Rome](#)

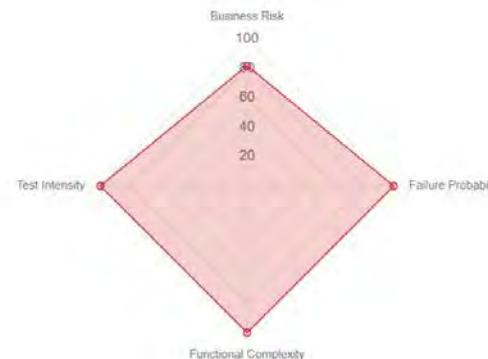
...

Description

Issue Panel

Add a description...

Journey To Rome DEV

[Restart Risk Assessment](#)**Custom field**

Risk evaluation DEV

Risk: A1

Complexity: HIGH

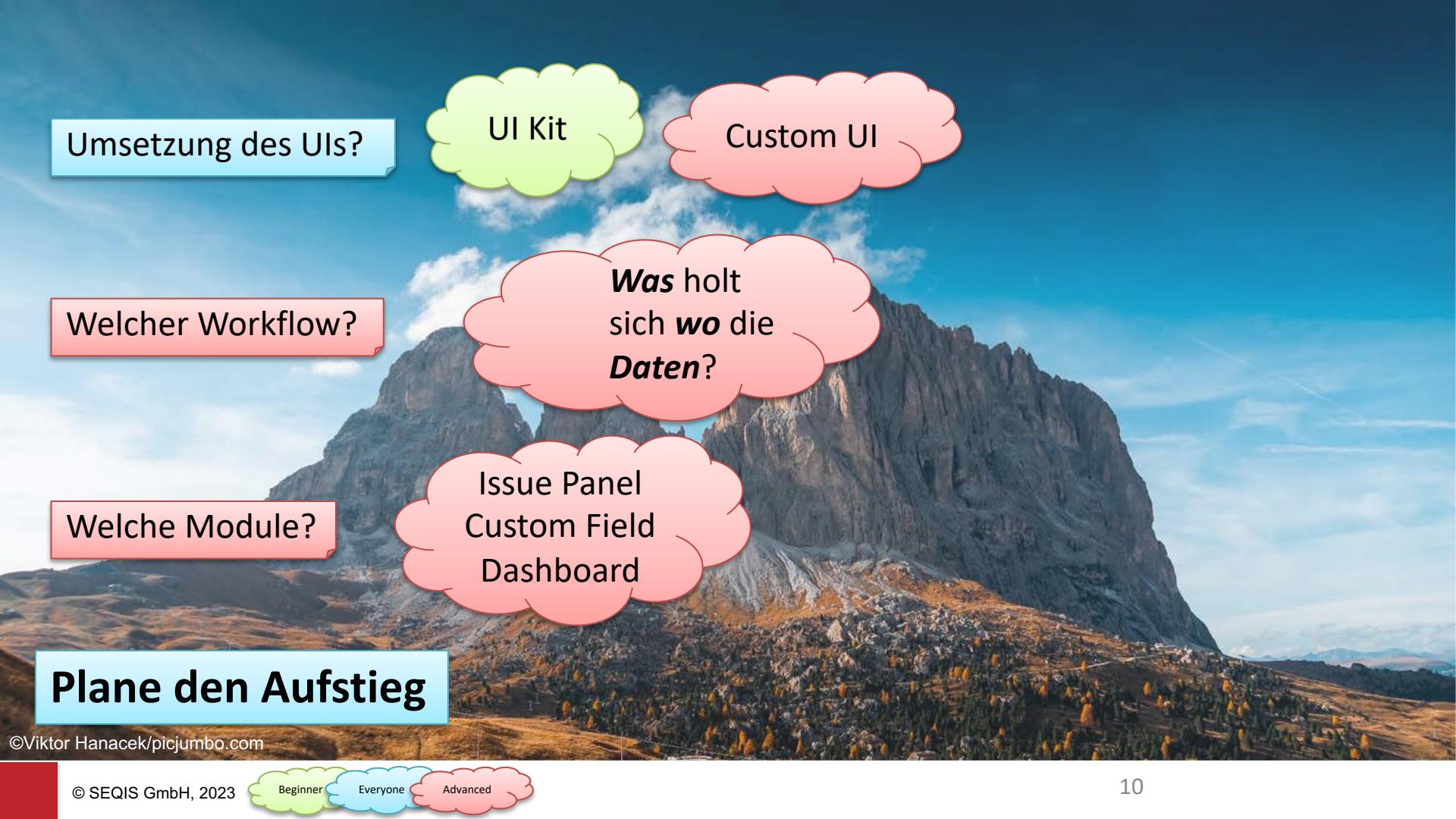
Test Intensity: 100

Values are computed

Priority

Medium

More fields Original esti...



Umsetzung des UIs?

UI Kit

Custom UI

Welcher Workflow?

Was holt
sich **wo** die
Daten?

Issue Panel
Custom Field
Dashboard

Welche Module?

Plane den Aufstieg

©Viktor Hanacek/picjumbo.com



1. Plane den Aufstieg deiner App in die Cloud

- Überlege dir, was die App wirklich können soll
- Prüfe, ob Forge für deine Idee geeignet ist
- Planen ist wichtig, Machen ebenso

One
more
thing

Anlegen

- Atlassian Partner Account
- API Token
- Atlassian Developer Account
- Atlassian Produkt Instanz

Node 16+



Docker



VS Code (o.ä.)



Voraussetzungen

©Engin_Akyurt/pixabay.com

Wichtig!

Dependencies
update!

1. Forge CLI installieren
2. Login mit Forge
3. App anlegen

```
1 npm install -g @forge/cli  
2  
3 forge login  
4  
5 forge create
```

Beispiel-Apps auswählbar
Guter Anhaltspunkt für Entwicklung

Einrichten

©Engin_Akyurt/pixabay.com

App bekommt
eigene ID!

1. App registrieren
 - Wenn schon vorhanden
2. App deployen
3. App installieren

Deploy und Install
nur für eigene App
möglich!

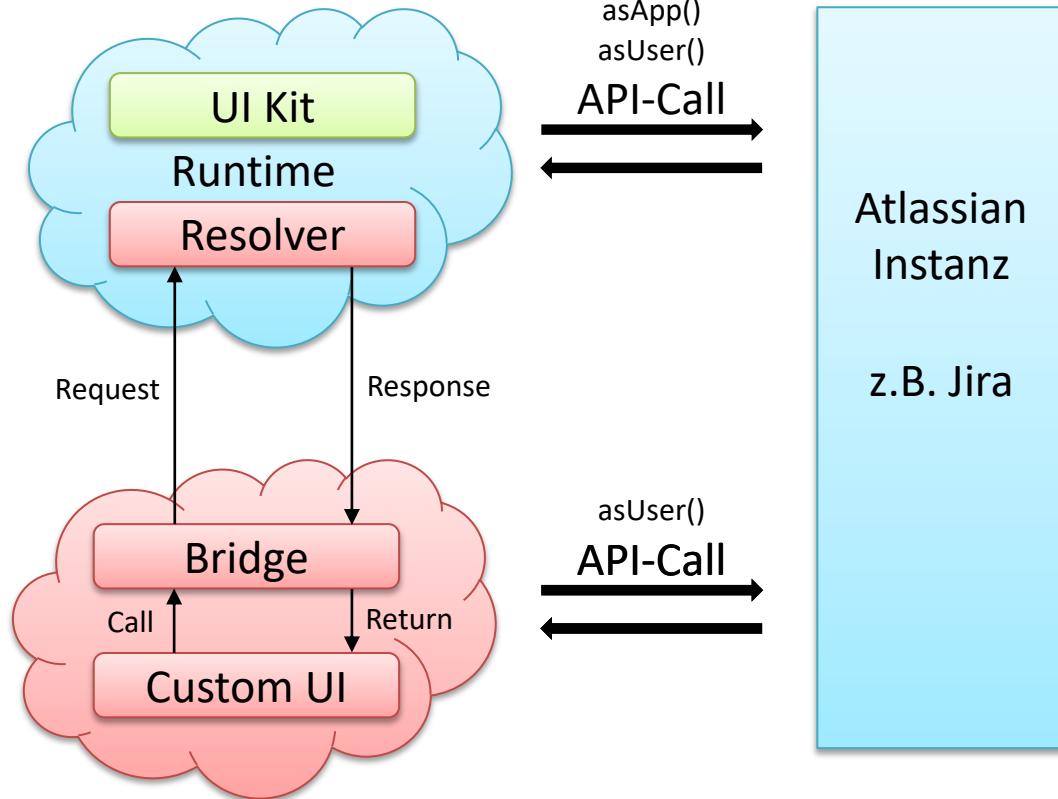
Deploy auf:
Develop
Staging
Production

```
1 forge register  
2  
3 forge deploy -e [environment]  
4  
5 forge install
```



App installieren

©Engin_Akyurt/pixabay.com



Die Forge Architektur



2. Die Forge-Architektur ist komplex

- Aufteilung in Runtime und Custom UI hilft bei der Planung
- Routine macht auch Komplexes alltäglich

One
more
thing

Custom UI Resolver

- Definition

Muss mit npm/yarn
installiert werden!

Fetch API

- Basic
- Product Aut.

Storage API

- Key –Value Sto
- Per Installation

API Limits beachten

!

Forge Runtime und API

```
const {resolver} = require('@forge/runtime');

const App = () => {
  return (
    <div>
      <h1>App</h1>
      <h2>User</h2>
      <h3>API-Call</h3>
    </div>
  );
};

module.exports = App;
```

```
'@forge/api';
"https://example.net/";

const App = () => {
  return (
    <div>
      <h1>App</h1>
      <h2>User</h2>
      <h3>API-Call</h3>
    </div>
  );
};

module.exports = App;
```

Scopes:
- storage:app

- Für „einfache“ App-Teile
- React-ähnlicher Syntax
- Produkt-spezifisch

UI Kit Hooks

- React Hooks

Einschränkungen

- Kein aktualisieren der View
- Forge Komponenten ≠ React Komponenten
- Schwer mit Unitests abdeckbar
- Nur eingeschränktes Set an UI-Elementen

```
1 import ForgeUI, { useProductContext } from "@forge/ui";
2
3 const platformContext = useProductContext().platformContext
```

Forge UI Kit

The screenshot shows a Microsoft Teams interface with a 'Risk Assessment' dialog open. The dialog contains fields for 'Risk' (set to 'A1'), 'Complexity' (set to 'High'), and 'Test Intensity (1-100)' (set to '100'). A red arrow points from the text 'Custom Field' to a callout box on the left side of the screen, which displays the calculated values: 'Risk evaluation: DEV', 'Risk: A1', 'Complexity: HIGH', 'Test Intensity: 100', and 'Values are computed'. Another red arrow points from the text 'Custom Field Edit' to the 'Edit values' section of the dialog, where the risk and complexity are being selected.

Custom Field

Custom Field Edit

Forge UI Kit

Debugging via Tunnel

1 forge tunnel

```
Listening for requests...
Serving file index.html for resource wizard
invocation: 9c64e60d85df41be risk: ...
INFO    07:56:19.466 9c64e60d85...
invocation: 983d425839303120 i...
invocation: a32e4791628665ec index...
```

Auch wenn App aktualisiert
werden soll!

	Name	Image	Status
	forge-tunnel-docker bf1737f12f2f8	atlassian/forge-tunnel:latest	Running



Größere Änderungen im Manifest

- Berechtigungen
- Module

2 forge deploy

3 forge install --upgrade

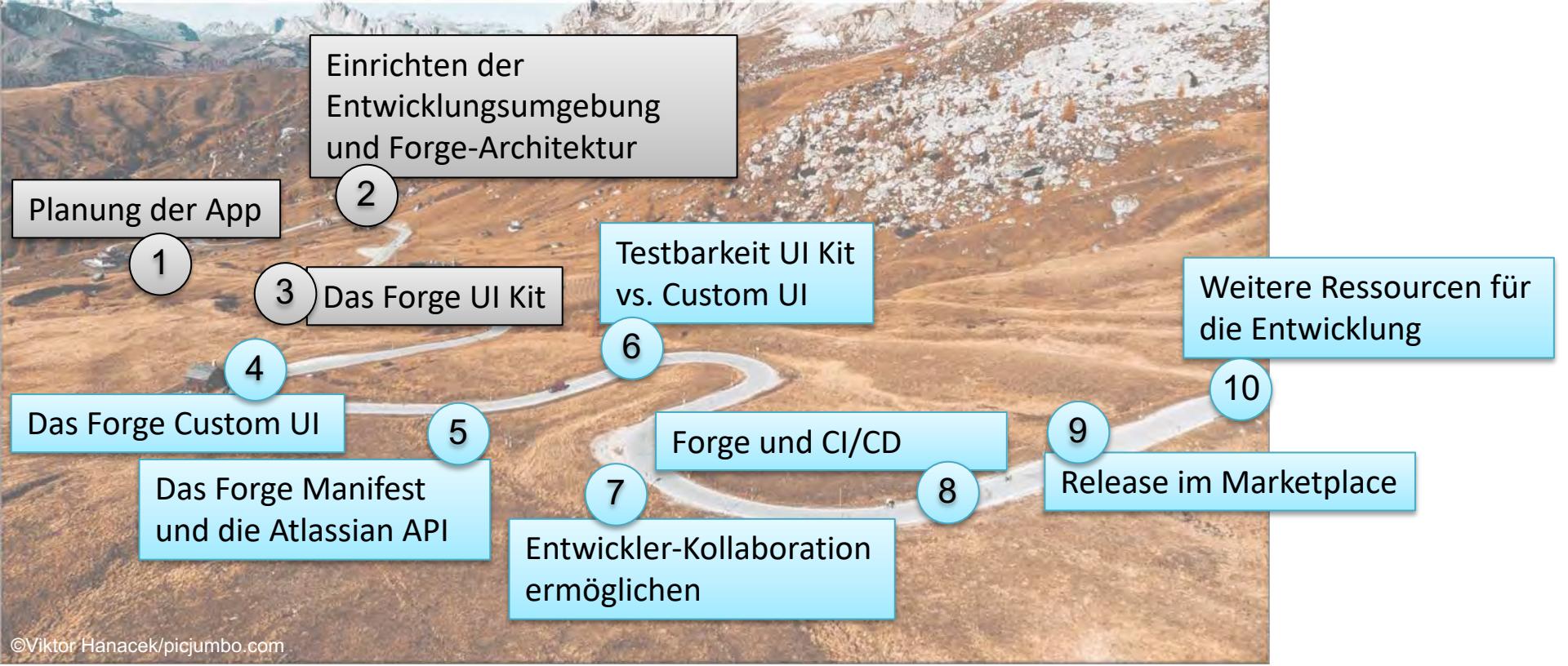
Development Workflow



3. Einfache Projekte? Verwende das UI Kit

- Ermöglicht dir einfache Apps zu erstellen
- Für komplexere Projekte ist ein Custom UI zu empfehlen
- Auch für erfahrenere Developer nützlich!

One
more
thing



Statische Resource via Iframe

Default Content Security Policy:

Externe Scripts/Assets

- External Bild
- External Script

Inline CSS



Einschränkungen

- Forge-API nur über Bridge-API
- „Hot Reload“ → Änderung im Manifest
- Kommunikation UI Kit  Custom UI

Custom Content Security Policy:

External Scripts/Assets via Bridge

```
1  'https://www.example.com'
2  'https://www.example-dev.com'
3  scripts:
4  - 'https://www.example-dev.com/script.js'
5
6  content:
7
8  styles:
9  - unsafe-inline
```

Forge Custom UI

API Limits beachten!

Custom UI Bridge API

Muss mit
installiert

```
1 import { invoke } from '@forge/bridge';
2
3 const getFoo = async () => await i
```

Muss mit
installiert

```
1 import { view } from '@forge/bridge';
2
3 await view.refresh();
4 const context = await view.getCon
```

nsApp()
 sUser()

l-Call

```
requestJira, requestConfluence } from '@forge/bridge';
  response = await requestJira('/rest/api/3/issue/ISSUE-1');
  confluenceResponse = await requestJira('/wiki/rest/api/content');
```

()

l-Call

Invocations

- Anzahl
- Laufzeit der Anfrage
- Returnierte Datenmenge

Ressource Updates

- Für Custom UI

Storage

- Kapazität
- Datenmenge Lesen/Schreiben

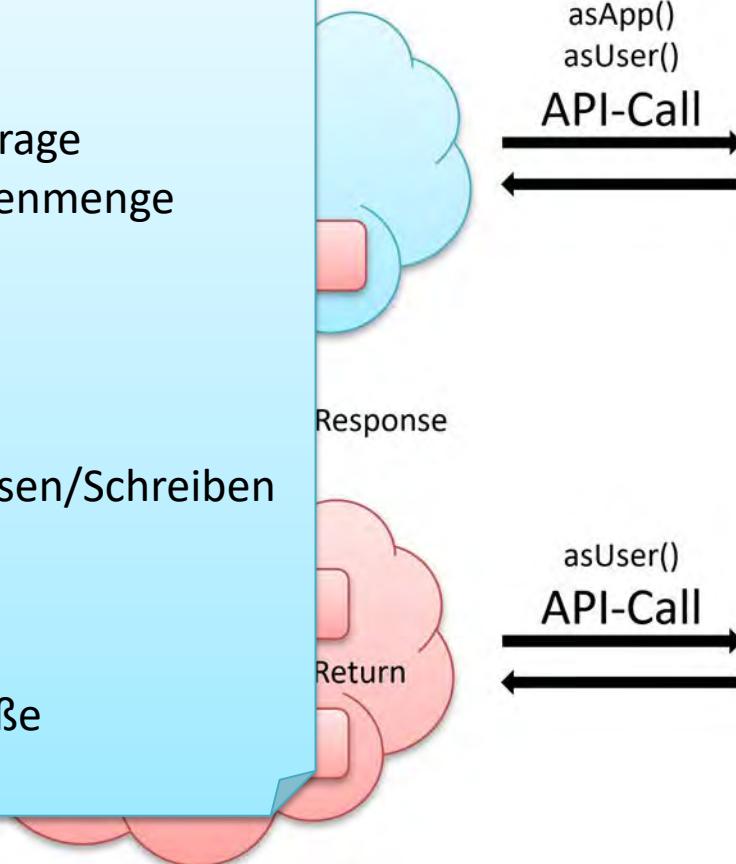
Memory

- App zu Laufzeit

Size

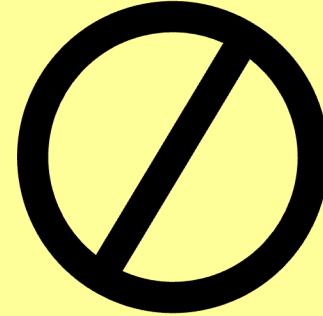
- App Gesamtgröße

Async Events (API)



Forge API Limits

Nicht abschrecken lassen!



Forge API Limits

View and filter app metrics to see how your app is performing. To learn more about each metric, see Monitor your apps.

All sites

Development

2023-03-30 10:00 ...

Reset

From Mar 30 06:00 (UTC) to Apr 12 06:00 (UTC) | Last updated a few seconds ago

Invocation success rate

100%

Overall

Invocation success rate: 99%

Percentage of successful invocations:

Group by: None

%

100

75

50

25

0

Invocation time

Time ranges showing speed of function invocations

Group by: None

%

100

75

50

25

0

<150ms

150-500ms

500-1000ms

1000-2000ms

2000-4000ms

>4000ms

Go to detailed view →

Noch nicht alle Limits
im Dashboard
Atlassian arbeitet daran

Forge API Limits

```
1 modules:  
2   jira:issuePanel:  
3     - key: example-app  
4       resource: wizard  
5       resolver:  
6         function: resolver  
7       title: Example App  
8       viewportSize: large  
9       tooltip: An app example
```

Runtime Resolver

asApp()

```
1 modules:  
2   function:  
3     - key: resolver  
4       handler: index.handler
```

```
1 resources:  
2   - key: wizard  
3     path: static/wizard/build
```

Bridge

asUser()

API-Call

Custom UI

Manifest Module

test2
Software project

PLANNING

TES Board
Board

Roadmap

Backlog

Active sprints

Reports

ISSUES

Components

DEVELOPMENT

Code

Releases

Project pages

Add shortcut

Projects / test2 / TES-4

test task

Attach Create subtask Link issue Journey To Rome

Description

Add a description...

Issue Panel

Journey To Rome DEV

Only UI

Calculation/
Validation

Data Change

What is affected by the required process/functionality?

- Only UI ⓘ
- Calculation Validation ⓘ
- Data change ⓘ

1 2 3 4 5 6 7 8 9 10 11 12

Abort

Next question

1

To Do Actions

Details

Assignee

Unassigned

Assign to me

Reporter

Daniel Kleissl

Labels

None

Risk evaluation DEV

Risk: A1

Complexity: HIGH

Test Intensity: 100

Values are computed

Priority

Medium

More fields

„Hot Reload“ → Eintrag im Manifest

- via React-Dev-Server

```
1 resources:  
2   - key: wizard  
3     path: static/wizard/build  
4     tunnel:  
5       port: 3000
```

Runtime

Resolver

```
1 forge deploy  
2  
3 forge install --upgrade
```

```
1 forge tunnel
```

Response

App aktualisieren

1. Build von Custom UI
2. Deploy via CLI

Bridge

```
1 react-scripts start
```

Custom UI

```
1 react-scripts build
```

Call

Development Workflow



4. Komplexe Projekte? Verwende das Custom UI

One
more
thing

- Du kannst alle Node Module nutzen, die React-kompatibel sind
- Fast alle Module unterstützen das Custom UI

Dashboard Modul

You are currently editing your dashboard. Changes will be saved automatically.

Journey to Rome dashboard (DEVELOPMENT)
By Daniel Kleissl
Dashboard gadget for the Journey to Rome plugin

	A	B	C
1	2	0	0
2	1	2	0
3	0	1	0

Selected Projects

- The best Project
- test2
- Test

Manifest

- Berechtigungen
- Ressourcen

Module

- Wo wird App dargestellt
- Zusätzliche Funktionen
- App-Informationen
- Icons

Manifest und Module

Relativ neu Bundle App+Dashboard

```
1 modules:  
2 jira:dashboardGadget:  
3   - key: example-dashboard-gadget  
4     title: Example Dashboard  
5     description: Example Dashboard  
6     thumbnail: https://developer.atlassian.com/platform/forge/images/icons/issue-panel-icon.svg  
7     resource: dashboard  
8     resolver:  
9       function: resolver  
10    edit:  
11      resource: dashboard  
12    function:  
13      - key: resolver  
14        handler: index.handler  
15  resources:  
16  - key: dashboard  
17    path: static/dashboard/build
```

Bug in Iframe



Workaround

```
1 document.documentElement.style.overflowY = 'visible';
```

Dashboard Modul

Abhängig von Atlassian Produkt
App-Berechtigungen im Manifest

Für Development

- „Classic“ Scopes verwenden
- Einschränken als letzter Schritt

Nach Installation muss Berechtigungen
zugestimmt werden

```
1 permissions:  
2   scopes:  
3     - read:field:jira  
4     - write:issue:jira  
5     - read:jira-user  
6     - read:jira-work
```

asApp()
asUser()

API-Call



Atlassian
Instanz
z.B. Jira

asUser()

API-Call



Atlassian Produkt API Berechtigungen



5. Beachte das Principle of Least Privilege

- Versuche, so wenig Berechtigungen wie möglich vom User anzufordern
- Keine Scheu vor Reverse Engineering bei Modulen

One
more
thing

UI KIT

Layout Tests
~~Layout Tests~~

Logic Tests nur von Forge-Modulen getrennt

Custom UI

Voraussetzung

- „React Testing Library“ (o.ä.)
- Mock-Framework z.B. Sinon.js

Vorbereitung

- Mock der Forge Bridge

Testen analog zu React

App testen

Custom UI

Muss vor jedem anderen File,
das @forge/bridge importiert,
gemocked werden!

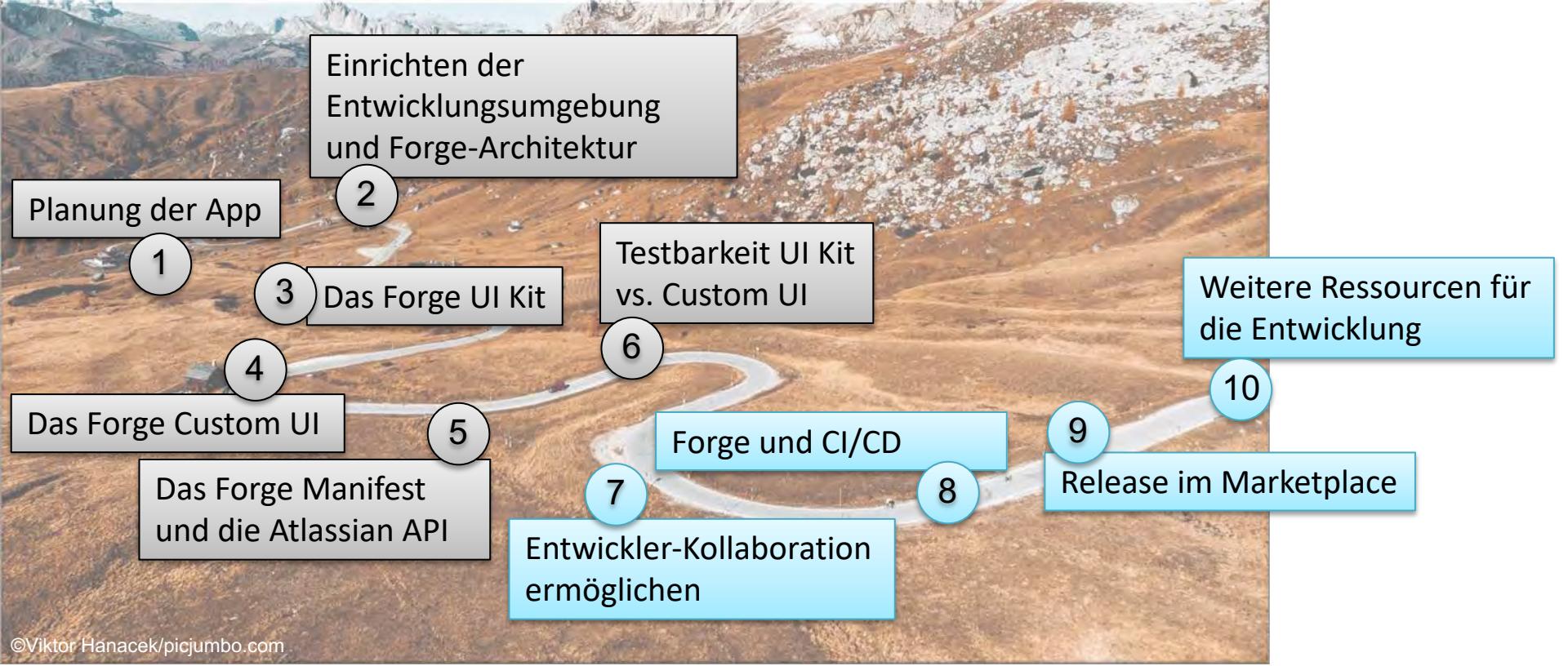
```
1 import sinon from 'sinon';
2
3 window.__bridge = {
4   callBridge: sinon.stub()
5 };
```



6. Testbarkeit wichtig? Verwende das Custom UI

One
more
thing

- UI Kit Layouts sind derzeit nicht testbar
- Nicht jeder Error ist auch euer Fehler



Herausforderung

Verknüpfung AppID und Dev-Account

Unsere Lösung

Ein Account pro Developer

- Atlassian Instanz
- App Registration

Ein „Master“-Account

- Produktiv-Umgebung
- Aktueller Release-Stand

AppID switching via Script

Atlassian arbeitet
an nativer Lösung!

1. AppID update
 - Setzt Developer AppID
2. Build und Deploy
 - Deploy auf Environment
3. AppID reset
 - Reset auf Master AppID

Developer Collaboration

©Andrii Yalanskyi/Shutterstock.com



7. Verwende mehrere Instanzen zur Kollaboration



- Ermöglicht deinem Team, unabhängig Features umzusetzen
- Definiere eine Instanz als Produktivumgebung um den Release Stand zu tracken

Herausforderung

- Installation ID nicht isoliert abfragbar
- *forge install* bei vorhandener Installation
- *forge uninstall* wenn keine Installation

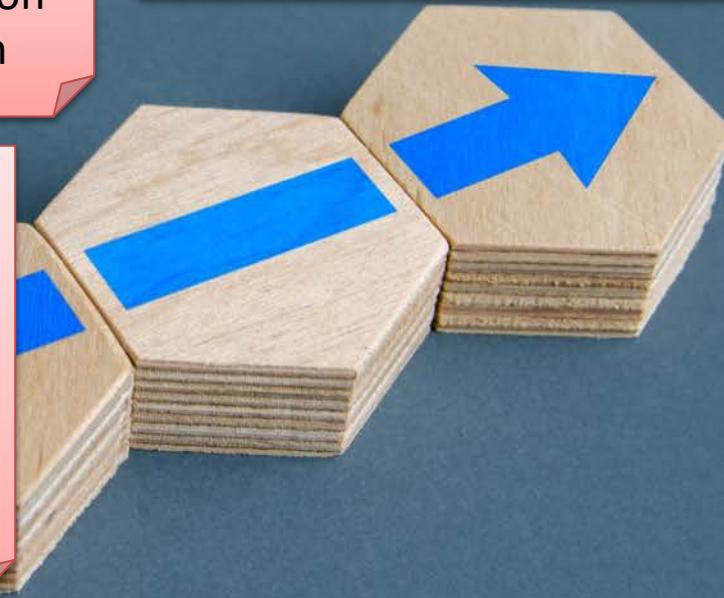
```
PS C:\Users\daniel.kleissl\Documents\razzfazz\repos\RnR\plugin> forge install list
Showing all the current installations of your app:

```

Installation ID	Environment	site
[REDACTED]	development	danielkleissl.atlassian.net

Unsere Lösung

1. Abfrage Installation ID
 - Via *grep* und *regex*
2. Deinstallation mit ID (wenn installiert)
 - *forge uninstall [Installation Id]*
3. Installation von neuer Version
 - *forge install*



Automation mit CI/CD

© Andrii Yalanskyi/Shutterstock.com



8. Forge App Deployment per CI/CD ist schwer, aber machbar

- Erleichtert verschiedene Umgebungen auf aktuellen Stand zu halten
- Behalte die Forge Roadmap im Auge

One
more
thing

Voraussetzung

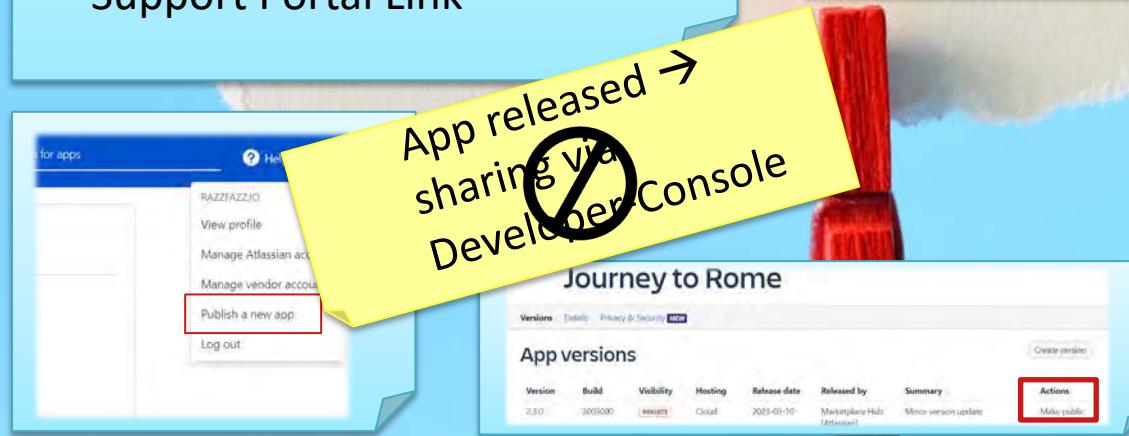
- App sharing aktiv
- Auf Production deployed
- Vendor Account
- Developer Community Account

Optional

- Security Self-Assessment
- Support Portal Link

Vor Release beantworten

- Gratis oder bezahlte Lizenz?
- App approval guidelines erfüllt?
- Welches Support Service Level Agreement?
- Welches end user license agreement (EULA)?
- Datenschutz-Grundverordnung eingehalten?
- Dokumentation vollständig und zugänglich?



Release im Marketplace

©Robert Plociennik/Shutterstock.com



9. Plane den Release im Marketplace gut

- Erstelle alle nötigen Marketing-Assets und Screenshots vor dem Release
- Du müsst die App nicht releasen um sie zu teilen

One
more
thing

ATASSIAN Developer Community

Interaktion mit Atlassian Experten
Einmelden von Bugs

ATASSIAN Developer

Blog
Aktuelle Issues
Changelog

ATASSIAN Marketplace

App Beispiele
EULA Beispiele von anderen Apps

Trello

Roadmap
Zukünftige Features

Zusätzliche Ressourcen

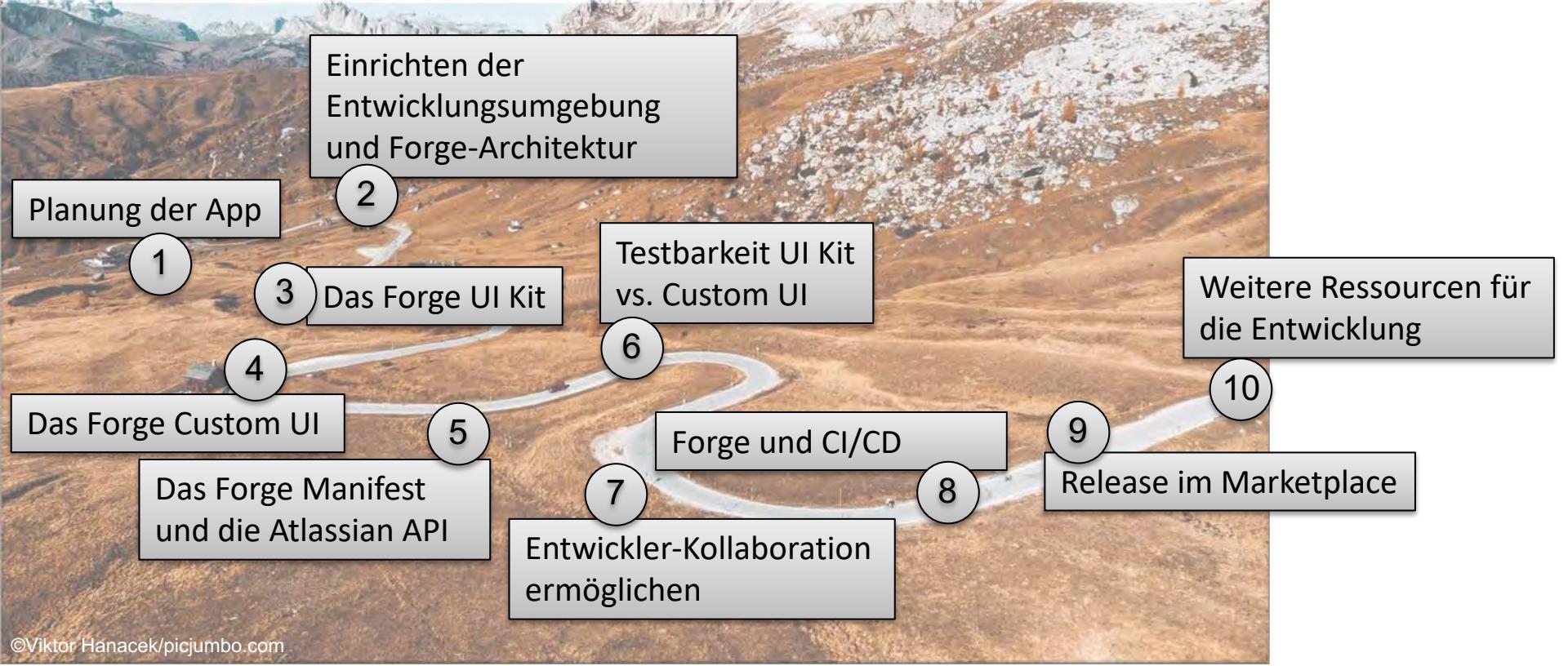
©Viktor Hanacek/picjumbo.com



10. Nutzt die Stärke der Community

- Die Atlassian Developer Community ist der beste Ort, um Hilfe bei Herausforderungen zu bekommen
- Bugs einmelden hilft!

One
more
thing



1. Plane den Aufstieg deiner App in die Cloud
2. Die Forge-Architektur ist komplex
3. Einfache Projekte? Verwende das UI Kit
4. Komplexe Projekte? Verwende das Custom UI
5. Beachte das Principle of Least Privilege
6. Testbarkeit wichtig? Verwende das Custom UI
7. Verwende mehrere Instanzen zur Kollaboration
8. Forge App Deployment per CI/CD ist schwer, aber machbar
9. Plane den Release im Marketplace gut
10. Nutzt die Stärke der Community