

# PROJECT COLECTIV

Károly Kovács & Otto Fischer

## Einführung

Stellt euch vor!

- Wie heißt du?
- Woher kommst du?
- Welche Programmiersprachen und Frameworks magst du bisher am meisten?
- Arbeitest du bereits in einem Unternehmen? In Teilzeit oder Vollzeit?



# Ziel

- Zusammenarbeit und Kommunikation üben
- SCRUM einsetzen
- Ein Projekt für euer Portfolio zu entwickeln

# Anforderungen

- Die Anwesenheit im Unterricht ist obligatorisch, und mehr als zwei Abwesenheiten führen zum Nichtbestehen.
- Es werden Projekte in Teams entwickelt, und jede Woche wird der Beitrag jeder einzelnen Person überprüft und benotet.
- Die Themen der Projekte sind vorgegeben, die Technologien werden im Team vereinbart.
- Die Umsetzung wird während des letzten Unterrichts als Team vorgestellt.

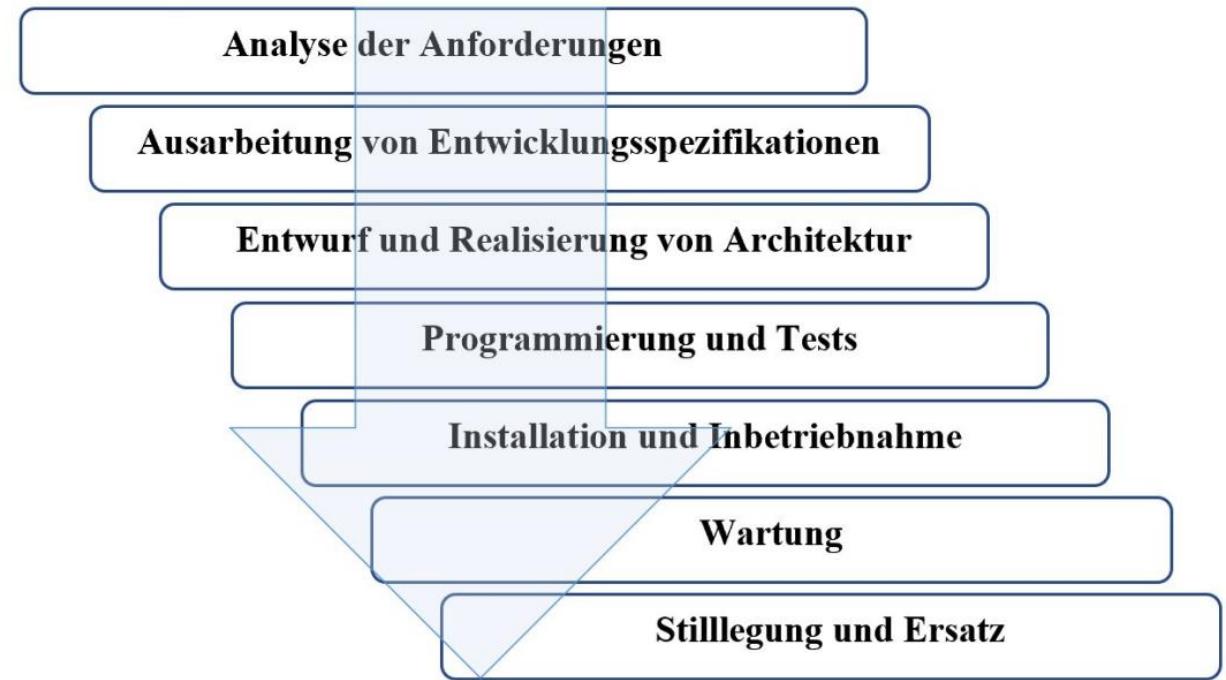
# Note

- 1 Punkt (offiziell) + 4 x 2 Punkt (Vorstellung des Vortschritts) + 1 Punkt für die Komplexität der Umsetzung

# Meilensteine

- Bis 30.10:
  - UML Diagramm und Use-Case Daigramme erstellen;
  - Datenbankstruktur erstellen;
  - Trello Board aufsetzen, alle Teammitglieder einladen, Epics definieren;
  - Technologien auswählen und Teamrollen definieren
- Bis 13.11:
  - Backend und Frontend aufsetzen
  - Registrierung- und Anmeldungs-Funktionalitäten
- Meilenstein 3 und 4 (bin Ende Dezember - Projektabhängige Bewertung)
  - Entwicklung der Funktionalitäten
- 13.01 und 15.01 Vorstellung der Projekten

- So wie Lebewesen geboren werden, wachsen, reif werden, altern und schließlich sterben, weisen auch kommerzielle Produkte ein ähnliches Verhalten auf. Sie werden entworfen, geplant, hergestellt, in Gebrauch genommen und irgendwann ersetzt, beziehungsweise ausgemustert.
- Die Zeitspanne vom Entwurf bis zur Markteinführung wird allgemein als Produktlebenszyklus („Product Life Cycle“) bezeichnet.
- Die Phasen des Lebenszyklus eines Softwareprodukts sind:
  1. Anforderungsanalyse;
  2. Erarbeitung der Entwicklungsspezifikationen;
  3. Entwurf und Erarbeitung der Architektur;
  4. Entwicklung/Programmierung und Tests;
  5. Installation und Inbetriebnahme;
  6. Wartung;
  7. Ausmusterung und Ersatz.



Was bedeutet ein erfolgreiches Projekt für euch?

Was sind die Hauptfaktoren für den Erfolg?

# Mögliche Faktoren

1. Idee, Geschäfts- und Prozessanalyse, Marktanalyse vom Fachbereich und BA
2. Vorplanung, Analyse der funktionalen und nichtfunktionalen Anforderungen
3. Zusammenstellung eines Teams mit hochaktuellen IT Kenntnisse
4. Kommunikation
5. Auswahl der richtigen Technologie
6. Auswahl des richtigen SDLC Entwicklungsmodells
7. Motivation und Identifikation mit dem Produkt

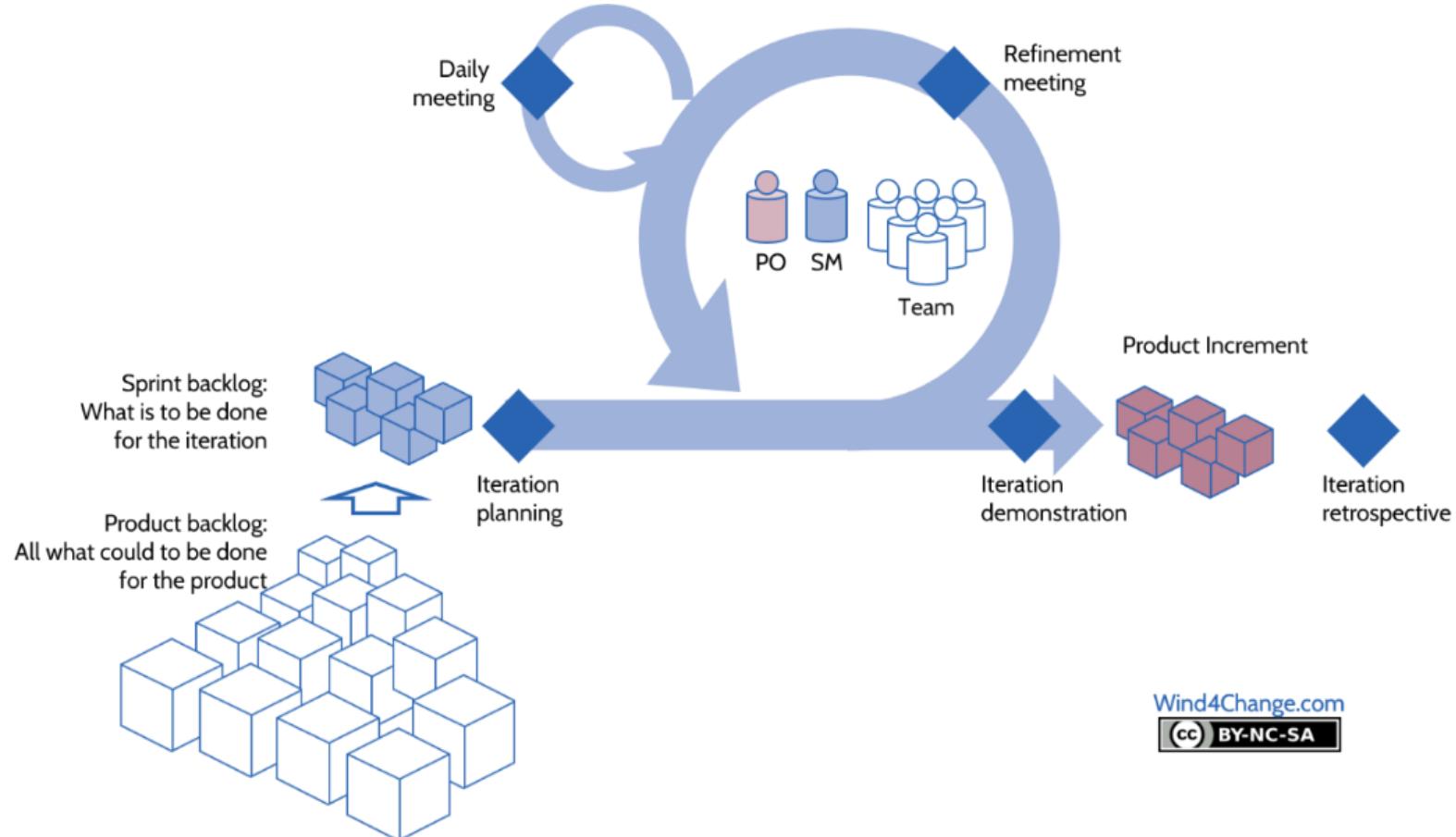
- „The choice of SDLC model is critical to the success of the type software project being implemented“
- Die Wahl des Modells für die Durchführung eines Softwareentwicklungsprojekts hängt von der Umsetzungsstrategie des Konzepts und den Grundsätzen des Lebenszyklus ab, wobei jedes Modell dieselben grundlegenden Schritte enthält, die gleichermaßen wichtig sind. Jedes Modell kann Anpassungen unterzogen werden, je nach dem konkreten Kontext.
- In der Praxis und der Fachliteratur sind derzeit mehr als 50 SDLC-Modelle bekannt. Die Wahl des optimalen SDLC-Modells hängt von mehreren Faktoren ab, darunter:
  1. Komplexität und Umfang der Anwendung;
  2. Verfügbarkeit aller Informationen von Anfang an, um die Entwicklungsanforderungen festzulegen;
  3. Mögliches Ausmaß der Interaktion mit dem Auftraggeber der Anwendung während des gesamten Projekts;
  4. Verfügbares Zeitbudget;
  5. Sicherstellung eines ressourcendeckenden Budgets;
  6. Akzeptables Risikoniveau;
  7. Ausmaß der angenommenen Flexibilität.

# Traditionelle lineare, sequenzielle SDLC-Modelle

- Traditionelle Softwareentwicklungsmodelle sind linear, sequenziell und basieren auf genau vordefinierten Phasen (Analyse, Anforderungsdefinition, Implementierung und Test, Wartung). Der Entwicklungsprozess verläuft unidirektional, wobei jede Phase so genannte Ergebnisse („Deliverables“) hat, die in validierten Dokumenten festgehalten werden. Diese Art von SDLC-Modellen wird angewendet, wenn die Anforderungen des Kunden von Anfang an klar sind und ein detailliertes Bild der zukünftigen Softwareanwendung vorliegt.
- Traditionelle Modelle sind von Natur aus zeit- und ressourcenaufwändig. Beispiele für lineare, sequenzielle SDLC-Modelle:
  1. Wasserfallmodell;
  2. V-förmiges Modell;
  3. Spiralmodell;
  4. Inkrementelles Modell

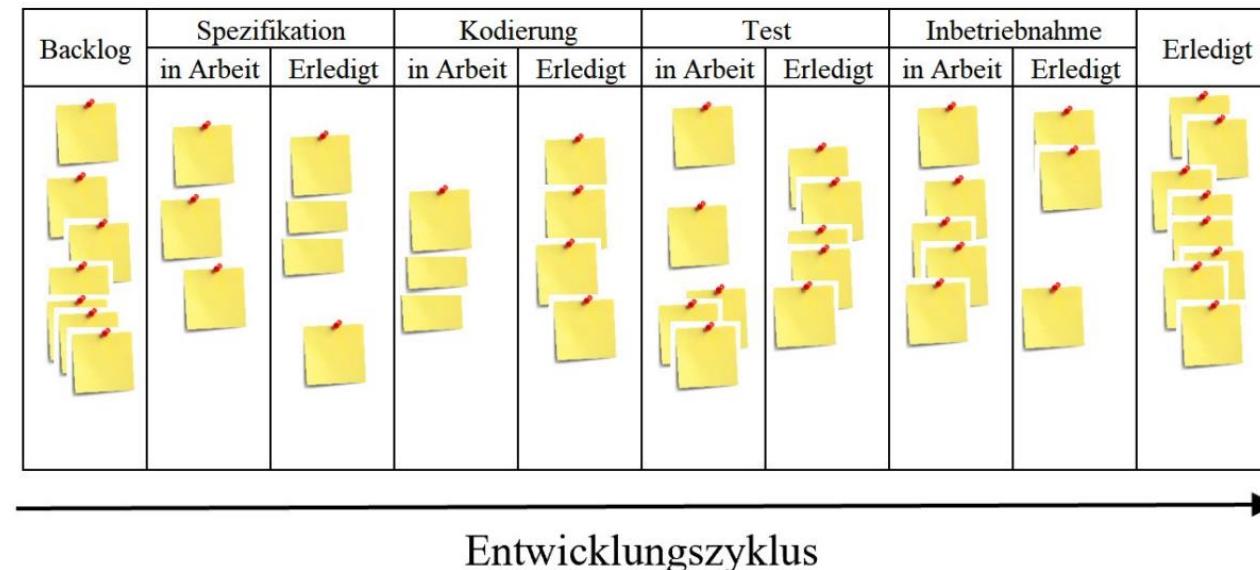
- Agile, iterative Modelle gehen nicht davon aus, dass alle Spezifikationen von Anfang an vorhanden sind. Man beginnt mit der Spezifikation und Implementierung eines Teils der Softwareanwendung. Der Prozess wird wiederholt, wobei nach jeder Iteration eine funktionierende Version entsteht, allerdings mit Teilfunktionalität.
- Jeder iterative Zyklus umfasst im Wesentlichen die typischen Phasen des traditionellen, linearen Modells, beginnend mit der Analyse, der Festlegung von Anforderungen, der Implementierung und dem Testen.
- Der „Schritt-für-Schritt“-Ansatz bietet die Möglichkeit, Funktionalitätspakete entsprechend den vom Auftraggeber gesetzten Prioritäten, aber auch entsprechend seiner Verfügbarkeit zu erstellen und zu überarbeiten, was zu einer intensiven Zusammenarbeit zwischen allen an dem Softwareentwicklungsprojekt beteiligten Akteuren führt.
- Bei innovativen Produkten, die zu Beginn bereits eine größere Komplexität haben, ist dieses Vorgehen alternativlos. Es sind pragmatisch Teilfunktionen zu entwickeln, die dann verifiziert und vor allem validiert werden. Auf die geschaffenen Ergebnisse können dann weitergehende, komplexe Funktionen realisiert werden. Iterative Modelle sind flexibel, effizient und kompatibel mit der dynamischen Entwicklung moderner Organisationen.
- Beispiele für agile, iterative SDLC-Modelle:
  1. Das „SCRUM“-Modell;
  2. Das „Kanban“-Modell;
  3. Viele Andere ....

- „SCRUM“ – eine SDLC-Methode, die auf iterativen und inkrementellen Prozessen basiert. Sie zeichnet sich durch Anpassungsfähigkeit, Flexibilität und Effektivität aus und wurde entwickelt, um dem Auftraggeber durch die Art und Weise, wie das Projekt zur Entwicklung von Softwareanwendungen durchgeführt wird, einen erheblichen Mehrwert zu bieten.
- Wir finden „SCRUM“ in Projekten, in denen die Definition der funktionalen Anforderungen schwierig und die Komplexität des Kontextes der Projektimplementierung hoch ist. Ein „SCRUM“-Projekt ist in „Sprints“ (Iterationen) von 2-4 Wochen unterteilt. Das Projektteam besteht aus maximal 10 Personen mit folgenden Rollen: „Scrum Master“, „Product Owner“ und „Entwickler“ („Developers“), die ohne hierarchische Unterordnung handeln, um ein sogenanntes Produktziel („Product Goal“) zu erreichen.
- Der „Scrum Master“ ist insbesondere für die Implementierung und Einhaltung der „SCRUM“-Methodik verantwortlich, aber auch für die Beseitigung möglicher Hindernisse, die während des Projekts auftreten.
- Der „Product Owner“ definiert das Produktziel und verwaltet das sogenannte „Product Backlog“, eine Liste aller Anforderungen („Benutzergeschichte“), die entwickelt werden müssen.
- Die Sitzungen während der Projektdurchführung haben spezifische Namen, wie z. B. „Daily Scrum“ oder „Sprint Retrospective“



# Das „Kanban“-Modell

- Das Kanban-Konzept wird heute in agilen Softwareentwicklungsprojekten zur Visualisierung, Planung und Lösung von Arbeitsaufgaben mit hoher Priorität aus der Liste der umzusetzenden Anforderungen („Backlog“) angewendet. Im Grunde handelt es sich um einen auf Tickets basierten Ansatz, bei dem Arbeitsaufgaben von einem Schild zum anderen verschoben werden, wobei der Entwickler, der eine Aufgabe am besten lösen kann, die Verantwortung für ihre Umsetzung übernimmt.
- Das Kanban-Modell ist in seiner Philosophie dem SCRUM-Modell sehr ähnlich. Wesentliche Stärke des Kanbansystems ist die hohe Dynamik. Die Zuständigkeiten werden nicht für einen langen Zeitraum zugeordnet. Je nach Verlauf und Notwendigkeit werden die „Ressourcen“ an die maximal effektiven Stellen genutzt.



# Projekt 1: Unternehmensplattform für interne Prozesse

Das System besteht aus mehreren Unternehmen. Jedes Unternehmen hat zwei Typen von Benutzern: Administratoren und Mitarbeitern. Jeder Benutzertyp hat sein eigenes Modul bzw. eine eigene Menge an Funktionalitäten.

## **Superadmin:**

1. Sie können neue Unternehmen einrichten mit mindestens einen Administrator.

## **Administratoren:**

1. Sie können sich über eine E-Mail-Adresse und ein Passwort ins System einloggen.
2. Sie können die Liste der Angestellten verwalten, die zum Unternehmen gehören.
3. Sie können für jeden Angestellten festlegen, wie viele Urlaubstage ihm laut Vertrag zustehen, und einen Prozentsatz definieren, der die erwartete Auslastung des Angestellten bestimmt.
4. Sie können eine Tabelle mit allen Urlaubsanträgen öffnen, in der sie die Anträge genehmigen oder ablehnen können. Sie können nur die Anträge sehen, mit denen sie verknüpft sind.
5. Sie können die Tätigkeitsberichte der Angestellten einsehen und prüfen, ob diese die Erwartungen erfüllt haben.

## **Mitarbeitern:**

1. Sie können sich über eine E-Mail-Adresse und ein Passwort ins System einloggen.
2. Sie können ein Dashboard öffnen, auf dem sie sehen, wie viele Urlaubstage sie im aktuellen Jahr noch übrig haben, und ihre bisherigen Urlaubsanträge tabellarisch einsehen. Alle Urlaubsanträge haben einen Status (Neu, Genehmigt, Abgelehnt) sowie eine generierte PDF-Datei mit Informationen zum Urlaubsantrag (Name, Start, Ende und Verlinkung zum Administrator, der die Genehmigung erteilt hat).
3. Sie können durch das Ausfüllen eines Formulars einen neuen Antrag stellen. Während dieses Prozesses wird ein PDF generiert und im System abgelegt.
4. Sie können einen Kalender öffnen, in dem sie für jeden Arbeitstag eintragen können, wie viele Stunden sie gearbeitet haben. Das System zeigt an, ob sie die Erwartungen der Administratoren erfüllt haben. (Hinweis: FullCalendar.js)

# Projekt 2: Digitalisierung einer Hochzeit

Das System unterstützt gleichzeitig mehrere Veranstaltungen. Eine Veranstaltung (z. B. eine Hochzeit) hat immer zwei Organisatoren und viele eingeladene Teilnehmer. Diese bilden die zwei Rollen mit unterschiedlichen Funktionalitäten.

## **Organisatoren:**

1. Sie können sich im System registrieren und eine Veranstaltung anlegen (Name, Start und Ende).
2. Sie können sich über eine E-Mail-Adresse und ein Passwort ins System einloggen.
3. Sie können für die erstellte Veranstaltung die Teilnehmerliste verwalten (CRUD + Import aus einer Excel-Datei).
4. Sie können per Knopfdruck auslösen, dass alle eingeladenen Personen eine Einladung per E-Mail erhalten.
5. Sie können den Ablauf der Veranstaltung definieren, der ebenfalls per E-Mail an alle Teilnehmer, die ihre Teilnahme bestätigt haben, verschickt werden kann.
6. Integration mit Google Maps

## **Teilnehmer:**

1. Sie können über den in der E-Mail erhaltenen Einladungslink ihre Teilnahme bestätigen oder ablehnen.
2. Bei der Bestätigung der Einladung können sie ein Konto erstellen, um sich später einloggen zu können.
3. Nach der Anmeldung sehen sie alle Veranstaltungen, zu denen sie zugesagt haben.
4. Sie können eine Veranstaltung öffnen und Fotos dazu hochladen. Alle Teilnehmer können die von anderen hochgeladenen Fotos durchstöbern.

# Projekt 3: ITSM Tool: Plattform für Ticketing und Bug-Meldungen

Das System besteht aus mehreren Unternehmen. Jedes Unternehmen hat zwei Benutzertypen: Administratoren und Entwickler. Jeder Benutzertyp hat sein eigenes Modul bzw. seine eigene Menge an Funktionalitäten.

## **Administratoren:**

1. Sie können sich und ihr Unternehmen registrieren und sich über eine E-Mail-Adresse und ein Passwort ins System einloggen.
2. Sie können Projekte verwalten (Name, Liste der Entwickler, Start- und Enddatum). Außerdem können sie Dateien zu den Projekten hochladen, wie z. B. Verträge, Dokumentationen, etc.
3. Sie können Kunden zum Projekt einladen, die dann eine Einladungs-E-Mail erhalten.
4. Sie können in der Projekt-Timeline neue Beiträge erstellen und kommentieren (ähnlich wie auf Facebook) sowie Bug-Tickets anlegen mit Name, Beschreibung, Priorität und Anhängen.

## **Kunden:**

1. Sie können einem Projekt über den Einladungslink beitreten.
2. Sie können die Projektinformationen einsehen, Dateien herunterladen, in der Projekt-Timeline neue Beiträge erstellen und kommentieren (ähnlich wie auf Facebook) sowie Bug-Tickets anlegen mit Name, Beschreibung, Priorität und Anhängen.

## **Entwickler:**

1. Sie können sich einloggen und die Projekte einsehen, mit denen sie verknüpft sind.
2. Sie können ein Projekt öffnen, sehen jedoch ausschließlich die Liste der Bug-Tickets.
3. Sie können sich Bug-Tickets zuweisen, diese als "Fertiggestellt" markieren und darüber hinaus wird der Administrator sowie der Kunde automatisch per E-Mail über die Fertigstellung eines Tickets benachrichtigt.

# Teams:

- Team 1:

- Anghel Mircea-Octavian
- Bandi Aron
- Chiorean Bogdan
- Cirlan George-Cosmin
- Cozac Raluca-Diana
- Draghici Eduard
- Ionescu David

- Team 2:

- Bacea Andreea-Maria
- Burlacu Robert-Adrin
- Chis Roxana-Maria
- Cosman Iulian
- Cracea Diana-Gabriela
- Gajdos-Rusu Radu-Gabriel
- Jitariu Rares-Stefan

- Team 3:

- Balc Felicia-Elena
- Chindris Raluca
- Ciobanu Sergiu-Tudor
- Coprean Vlad-Catalin
- Pop David-Mihai
- Goc Raluca-Maria
- Lemeni Iris-Alma

Benötigte Rollen pro Team:

- Team lead
- Scrum master

# Herzlichen Dank für Ihre Aufmerksamkeit!

Károly Kovács & Otto Fischer

**MHP Management- und IT-Beratung GmbH**

Film- und Medienzentrum | Königsallee 49 | D-71638 Ludwigsburg  
Telefon +49 (0)7141 7856-0 | Fax +49 (0)7141 7856-199  
eMail [info@mhp.com](mailto:info@mhp.com) | Internet [www.mhp.com](http://www.mhp.com)