



Ogrodje Scrum v pedagoškem procesu: način agilne organizacije skupinskega dela

Oddelek za izobraževanje in študij

Junij, 2026

Kazalo vsebine

1	Uvod v ogrodje Scrum	2
2	Osnovna načela Scrum	3
3	Vloge v Scrum Team	4
4	Potek dela po ogrodju Scrum	7
5	Scrum artefakti	10
6	Uporaba Scrum v visokoškolskem izobraževanju	11
6.1	eduScrum kot pedagoška prilagoditev Scrum	14
7	Primeri uporabe v izobraževanju	17
8	Viri	21

1 Uvod v ogrodje Scrum

Scrum je način organizacije dela v skupini, pri katerem multidisciplinarna ekipa projekt razvija postopno, v kratkih ciklih (t. i. **sprintih**), ob rednem preverjanju rezultatov in sprotne prilagajanju nadaljnega dela. Scrum se je izvirno pojavil na področju razvoja programske opreme, danes pa se uspešno uporablja tudi že na drugih področjih.

Osnovna ideja Scrum je **iterativno** reševanje kompleksnih nalog. Ekipa v vsakem sprintu razvije del rešitve, po zaključku sprinta pregleda rezultate, pridobi povratne informacije in na tej osnovi prilagodi nadaljnje delo. Za razliko od klasičnih projektnih pristopov Scrum ne zahteva, da je projekt v celoti načrtovan vnaprej. Namesto tega omogoča testiranje rešitev in prilagajanje med samim procesom dela.

Scrum je del širše skupine t. i. **agilnih pristopov** (angl. *agile*), ki poudarjajo tovrstno iterativno delo, sodelovanje in sprotne prilagajanje razvoja. V praksi se poleg Scrum uporabljajo tudi drugi pristopi za organizacijo dela, na primer metoda Kanban, ki izhaja iz **vitkih pristopov** (angl. *lean*). Pri Kanban so delovni elementi vizualizirani, kar omogoča enostavnejše upravljanje poteka dela in spremljanje napredka nalog. Pogosto se uporablja tabla, na kateri so naloge razporejene po fazah dela, na primer »*To do*« (naloge, ki še niso začete), »*In progress*« (naloge v teku) in »*Done*« (opravljene naloge). Čeprav Kanban ni del ogrodja Scrum, se v praksi pogosto uporablja skupaj s Scrum za pregled nad nalogami.

Potek dela po Scrum, kot je opisan v nadaljevanju, je povzet po [vodiču Scrum](#), ki sta ga objavila Jeff Sutherland in Ken Schwaber. Vodič sta v slovenščino prevedla Voranc Kotnik in Matej Golob in tudi prevod je, tako kot original, brezplačno dosegljiv na spletu ([prevod vodiča](#)). Ker je v tem gradivu govora o izvornem Scrum, ključni izrazi niso prevedeni v slovenščino (kot velja tudi za vodič). Ob njih so podana pojasnila, ki so pedagoškim izvajalcem lahko v pomoč pri vključevanju pristopa v pedagoški proces.

Spoznavanje Scrum v visokošolskem izobraževanju lahko študentom pomaga razvijati kompetence, ki so pomembne za projektno organizirano delovno okolje. Scrum in njegova uporaba (lahko tudi zgolj posameznih elementov) v pedagoškem procesu spodbuja aktivno in sodelovalno učenje, razvija kompetence timskega dela, kot so učinkovita komunikacija, organizacija dela, skupno sprejemanje odločitev in reševanje problemov. Hkrati spodbuja samostojnost in prevzemanje odgovornosti posameznika za skupne rezultate ter refleksijo.

V pedagoški proces ni treba vključiti Scrum ogrodja v celoti. Izvajalci lahko vključijo posamezne elemente, na primer kratke razvojne cikle, redna srečanja študentov za usklajevanje dela ali refleksijo. Scrum je še posebej primeren pri projektno ali problemsko zasnovanem učenju, kjer študenti v skupinah rešujejo kompleksnejše naloge, ustvarjajo izdelke ipd.

Tudi v delovnem okolju Scrum pogosto ni uporabljen v povsem izvorni obliki, temveč obstajajo prilagoditve, ki se prilagajajo značilnostim projekta ali delovnega okolja.

2 Osnovna načela Scrum

Scrum temelji na **treh ključnih načelih empirizma**:

- transparentnost (*Transparency*) – vsi člani ekipe imajo vpogled v stanje projekta
- pregled (*Inspection*) – ekipa redno preverja napredek in rezultate
- prilagodljivost (*Adaptation*) – delo in projekt se na podlagi ugotovitev sproti prilagaja

Ta načela omogočajo ekipi, da se učinkovito odziva na spremembe in postopoma izboljšuje rešitve.

Preslikava na pedagoški proces:

Skupina študentov pri projektne delu razvija prototip mobilne aplikacije in usklajuje delo in naloge z uporabo spletnega orodja, kjer je ves čas vsem na voljo vpogled v trenutno stanje in celotno dogajanje (*Transparency*). Po vsakem krajšem ciklu dela izvajalec skupaj s študenti pregleda rezultate (*Inspection*), študenti dobijo povratne informacije in prilagodijo nadaljnje delo (*Adaptation*) za nov cikel skupinskega dela.

Poleg načel Scrum poudarja še **pet temeljnih vrednot**:

- zavezanost (*Commitment*),
- osredotočenost (*Focus*),
- odprtost (*Openness*),
- spoštovanje (*Respect*),
- pogum (*Courage*).

Te vrednote spodbujajo **učinkovito sodelovanje znotraj ekipe**.

Preslikava na pedagoški proces:

- zavezanost: vsi študenti v skupini prevzamejo odgovornost za skupni cilj
- osredotočenost: študenti se osredotočajo na (svoje) naloge sprinta
- odprtost: študenti odprto delijo informacije, poročajo o svojem napredku in tudi o morebitnih težavah
- spoštovanje: študenti spoštujejo prispevek vseh članov v skupini
- pogum: vsi člani skupine si upajo opozoriti na težave, vprašati za pomoč, preizkusiti nove pristope

3 Vloge v Scrum Team

Scrum Team je osnovna delovna enota. Gre za multidisciplinarno ekipo, ki dela na doseganju skupnega cilja projekta (**Product Goal**). Znotraj ekipe ni dodatnih podskupin in formalne hierarhije. Člani so izbrani tako, da skupno prispevajo vse potrebne kompetence za razvoj rešitve oziroma želenega produkta.

Za Scrum Team je zelo značilno **samoupravljanje** (*self-management*), kar pomeni, da člani sami organizirajo svoje delo in dogovorijo, kdo bo določeno nalogo izvedel, kdaj in na kakšen način. Scrum Team je običajno majhen (do deset članov), da ostane učinkovit in prilagodljiv.

Znotraj Scrum Team obstajajo **tri ključne odgovornosti** oziroma **vloge**.

Product Owner: predstavlja povezavo med različnimi deležniki projekta (*stakeholders*) in ekipo razvijalcev. Deležniki so lahko na primer končni uporabniki, naročniki, investitorji. Product Owner poleg tega skrbi tudi za to, da ekipa dela na nalogah, ki prinašajo največjo vrednost oziroma ustvarjanje napredka.

Njegove naloge so:

- razvijanje in jasno komuniciranje končnega cilja (**Product Goal**),
- razumevanje potreb deležnikov in upravljanje seznama nalog (**Product Backlog**),
- razvrščanje nalog po prioriteti.

Primer: Predpostavimo, da študenti pri učni enoti razvijajo prototip mobilne aplikacije za obveščanje o študentskih dogodkih.

V tem primeru so deležniki:

- študenti, ki bodo aplikacijo uporabljali za spremljanje dogodkov,
- organizatorji dogodkov (npr. študentski svet, tutorji, fakulteta), ki želijo dogodke objavljati in promovirati,
- visokošolski učitelj, ki usmerja projekt in spremlja doseganje učnih ciljev,
- fakulteta, če želi izboljšati obveščanje študentov in njihovo vključenost v obštudijske dejavnosti.

Naloga vloge Product Owner je, da potrebe deležnikov pretvori v konkretne naloge te študentske skupine, na primer:

- prijava na dogodek,
- obvestila o novih dogodkih,
- filtriranje dogodkov po kategorijah.

Scrum Master: skrbi za pravilno uporabo Scrum ogrodja pri delu ekipe. Pomembno je poudariti, da Scrum Master ni vodja ekipe.

Njegove naloge vključujejo:

- zagotavljanje, da celotna ekipa razume Scrum pristop,
- odstranjevanje ovir, ki preprečujejo napredek pri delu ekipe,
- spodbujanje učinkovite komunikacije oziroma sodelovanja (podpora pri samoupravljanju ekipe).

Primer: Pri prej omenjenem študentskem projektu lahko vlogo Scrum Master prevzame eden izmed študentov v skupini.

Študent, ki prevzame to vlogo:

- organizira kratke sestanke skupine (**Daily Scrum**),
- skrbi, da skupina sledi načrtu v posameznem ciklu,
- pomaga skupini rešiti organizacijske težave (npr. nejasna naloga ali slaba komunikacija),
- vodi srečanje namenjeno refleksiji (**Sprint Retrospective**), kjer skupina razmisli, kako izboljšati delo v naslednjem ciklu.

Pri tem študent, ki prevzame vlogo Scrum Master, nikoli ne določa nalog drugim študentom, ampak spodbuja skupino, da se sama dogovori, kako bo delo organizirano. Pomembno je tudi, da dobro pozna Scrum. Vlogo lahko prevzame tudi izvajalec.

Developers: so člani ekipe, ki izvajajo projektno delo in razvijajo rešitev oz. produkt (**Product**). V vsakem sprintu so odgovorni za **Increment**, kar lahko razložimo kot prirastek vrednosti oz. nov napredek h končni rešitvi oz. rezultatu.

Odgovorni so na primer za:

- izdelavo načrta za vsak sprint (**Sprint Backlog**),
- zavezanost prevzemanju odgovornosti in zagotavljanje kakovosti dela tako, da zadostijo **Definition of Done** (predstavlja dogovorjene kriterije, ki določajo, kdaj je rezultat dela dovolj kakovosten, da se šteje kot dokončan),
- vsakodnevno prilagajanje načrta v smeri cilja (**Sprint Goal**).

Primer: V pedagoškem kontekstu praktičnega primera vlogo Developers prevzamejo študenti v projektni skupini. Njihova naloga je načrtovanje nalog v posameznih ciklih dela, izvajanje projektnega dela ter priprava delnih prirastov pričakovanega rezultata.

Njihove aktivnosti lahko vključujejo na primer:

- oblikovanje koncepta mobilne aplikacije,
- pripravo uporabniškega vmesnika,
- razvoj ali prototip funkcionalnosti,
- testiranje in izboljševanje rešitve,
- pripravo predstavitve rezultatov.

Študentska skupina v vsakem ciklu razvije del rešitve, na primer:

- prototip koledarja dogodkov,
- funkcijo za prijavo na dogodek,
- sistem obveščanja uporabnikov.

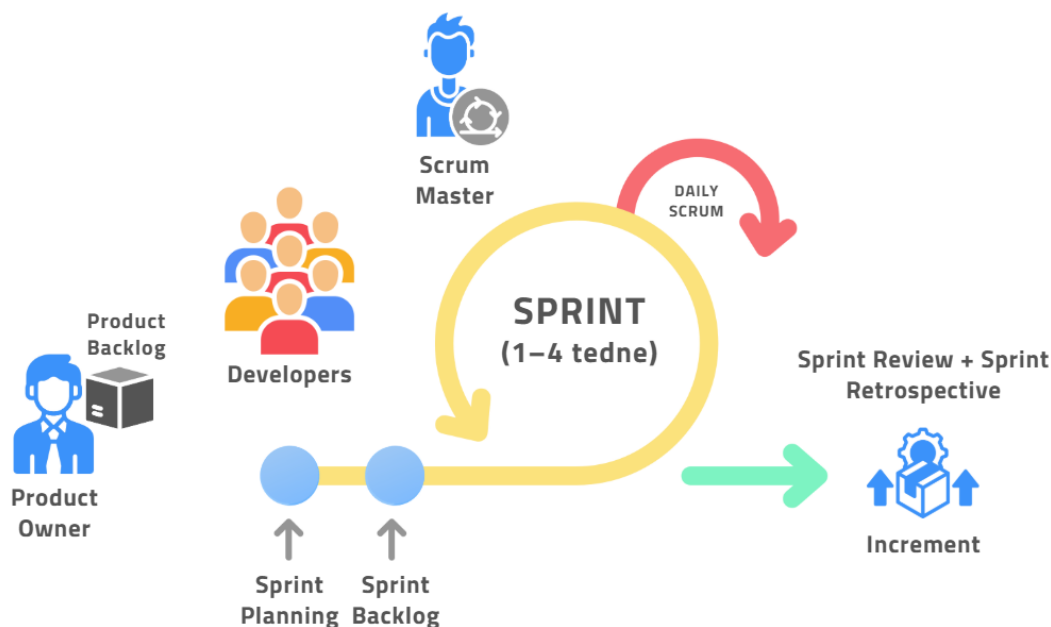
V tabeli 1 so povzete ključne informacije o vlogah, ki so prisotne v Scrum in preslikava le-teh v pedagoški proces.

Tabela 1: Vloge v Scrum Teamu in preslikava na pedagoški proces

Vloga:	Osnovna odgovornost:	Komu ustreza vloga v pedagoškem procesu:
Product Owner (produktni vodja)	določa vizijo produkta in prioritete nalog	Izvajalec učne enote, ki ima pregled nad učnimi cilji in lahko lažje usmerja projekt. Študent (vodja skupine).
Scrum Master (vodja metodologije)	skrbi za učinkovit potek delovnega procesa (po principih Scrum) in učinkovito sodelovanje ekipe	Izvajalec ali študent, ki dobro pozna Scrum in podpira študente pri reševanju organizacijskih in drugih težav, spodbuja skupino k sodelovanju in samoupravljanju, skrbi za izvedbo refleksije.
Developers (razvijalci)	izvajajo projektno delo	Skupina študentov, ki razvija rešitev.

4 Potek dela po ogrodju Scrum

Delo v Scrum poteka v iteracijah, imenovanih **sprinti** (slika 1). Sprint je časovno omejen cikel, ki običajno traja 1–4 tedne, najpogosteje pa dva tedna. Vsak sprint vključuje več ključnih korakov oz. dogodkov, ki so podrobneje opisani v nadaljevanju.



Slika 1: Scrum ogrodje

1. *Sprint Planning* (pomeni načrtovanje sprinta)

Na začetku sprinta ekipa določi:

- katere naloge bodo izvedene,
- kakšen je cilj sprinta (*Sprint Goal*).

Naloge za sprint (***Sprint Backlog***) se izberejo iz seznama vseh nalog (***Product Backlog***).

2. *Daily Scrum* (predstavlja kratek dnevni sestanek)

Ekipa lahko izvede dnevni sestanek ekipe (običajno do 15 minut), na katerem člani ekipe:

- pregledajo napredek,
- identificirajo morebitne ovire,
- uskladijo nadaljnje korake.

Daily Scrum spodbuja in krepi komunikacijo, podpira hitro odločanje in posledično zmanjša potrebo po dodatnih sestankih. Zanimivo je, da se tak dnevni sestanek zaradi poenostavitve običajno izvede vedno ob istem času in na istem mestu.

3. *Sprint Review* (pomeni pregled rezultatov sprinta)

Na koncu vsakega sprinta ekipa predstavi rezultate:

- kaj je bilo razvito,
- kaj deluje,
- kakšne povratne informacije imajo deležniki.

Ob pregledu napredka se razpravlja o tem, kako naprej. Zato je to delovno srečanje in ne zgolj predstavitev. V primeru enomesečnega sprinta, je trajanje omejeno na največ štiri ure. Pri krajših sprintih pa je to srečanje ustrezno krajše.

4. *Sprint Retrospective* (predstavlja refleksijo o delu)

Retrospektiva je refleksija o procesu dela. Njen namen je določiti načine za izboljšavo kakovosti in učinkovitosti dela Scrum ekipe.

Ekipa razmisli:

- kaj je delovalo dobro,
- kaj jo je oviralo,
- kaj bi lahko izboljšali,
- kako organizirati delo v naslednjem sprintu.

Retrospektiva zaključí posamezni sprint.

Preslikava Scrum dogodkov v pedagoški proces:

Tabela 2: Scrum dogodki pri študentskem projektu

Scrum dogodek	Kaj pomeni v Scrum procesu	Primer v pedagoškem procesu
1. <i>Sprint Planning</i>	Ekipa na začetku Sprinta določi, katere naloge iz Product Backlog bo izvedla in s tem opredeli Sprint Backlog. Določi se, kaj bo Sprint Goal.	Študentska skupina pregleda seznam nalog projekta in se odloči, katere naloge bo izvedla v naslednjih dveh tednih (npr. razvoj koledarja dogodkov v aplikaciji).
2. <i>Daily Scrum</i>	Kratek dnevni sestanek (15 minut) ekipe za pregled napredka, identifikacijo ovir in usklajevanje nadaljnjih korakov.	Na začetku srečanja skupine vsak študent na kratko pove, kaj je naredil, kaj bo naredil v nadaljevanju in ali se je pojavila kakšna težava. Ker študenti pogosto nimajo dnevnih srečanj, lahko usklajevalni sestanek poteka 1–2-krat na teden.
3. <i>Sprint Review</i>	Ekipa predstavi rezultate Sprinta in pridobi povratne informacije deležnikov. V primeru enomesečnega sprinta	Študentska skupina predstavi delujoč prototip funkcionalnosti aplikacije (npr. pregled dogodkov) ter prejme povratne informacije izvajalca in študijskih kolegov iz drugih skupin. Sprint Review je lahko

	so to največ štiri ure, pri krajših sprintih pa ustrezno manj.	priložnost za sprotno povratno informacijo učitelja. Trajanje je odvisno od obsega in trajanja samega sprinta. Dodatne okoliščine so lahko še število študentskih skupin, zato izvajalec trajanje ustrezno načrtuje (npr. 20–45 minut).
4. Sprint Retrospective	Ekipa reflektira proces dela in razmisli, kako izboljšati sodelovanje v naslednjem Sprintu.	Študenti skupaj razpravljajo, kaj je v skupini delovalo dobro, katere težave so se pojavile, kako jih odpraviti in kako bi lahko delo v naslednjem sprintu organizirali še bolj učinkovito.

5 Scrum artefakti

Koncept artefaktov je pogosto najtežje razumljiv začetnikom. Artefakti pomagajo zagotavljati preglednost napredka in omogočajo skupini, da se osredotoči na skupni cilj projekta. Scrum uporablja **tri ključne artefakte**, ki predstavljajo delo ali vrednost. Predstavljeni so v tabeli 3 skupaj z možno preslikavo na pedagoški proces.

Tabela 3: Artefakti in preslikava na pedagoški proces

Artefakt	Opis	V kontekstu pedagoškega procesa
Product Backlog	Urejen seznam vseh nalog, ki so potrebne za doseg cilja (razvoj produkta). Seznam je običajno urejen po nujnosti opravljanja nalog. Zaveza artefakta je Product Goal .	Seznam vseh projektnih nalog skupine študentov.
Sprint Backlog	Je načrt dela oz. seznam nalog določen za posamezni sprint, ki jih je ekipa izbrala iz Product Backlog. Zaveza artefakta je Sprint Goal .	Naloge za 1–2 tedna projektne delo študentov.
Increment	Predstavlja konkretni rezultat sprints, prirast oz. napredek produkta, ki je dokončan, in mora zadostiti dogovorjenim kriterijem kakovosti (Definition of Done). Idealen napredek pomeni zaključen izdelek.	Ustvarjen napredek na projektu po dveh tednih, ki zadošča kriterijem uspešnosti.

Primer: Študenti pri učni enoti razvijajo prototip mobilne aplikacije za obveščanje o študentskih dogodkih. Na začetku projekta skupina skupaj z izvajalcem pripravi seznam nalog (*Product Backlog*), s katerimi bodo ustvarili končni izdelek (prototip aplikacije). Ker skupina vseh nalog ne more izvesti naenkrat, na začetku sprints iz seznama izbere smiselno količino nalog, ki jih bo izvedla v prvem obdobju. Te naloge tvorijo *Sprint Backlog*, ki predstavlja načrt dela skupine za ta sprint.

Na primer, skupina se odloči, da bo razvila:

- osnovni uporabniški vmesnik aplikacije,
- osnovno strukturo podatkov za dogodke,
- pregled dogodkov v obliki seznama.

Po zaključku sprints skupina predstavi prvi delni rezultat in pridobi povratne informacije. Ta rezultat predstavlja napredek (*Increment*), torej konkreten del aplikacije. V nadaljevanju bi skupina izvedla še retrospektivo o poteku dela. Nato pa načrtala delo za naslednje obdobje (sprint) z novim izborom nalog in več izkušnjami za učinkovito sodelovanje. Pri tem je aktivno vključen tudi izvajalec, ki usmerja pogovor retrospektive, opozarja na kritične točke in priložnosti za izboljšave, ki jih študenti sami morda ne prepoznajo.

6 Uporaba Scrum v visokošolskem izobraževanju

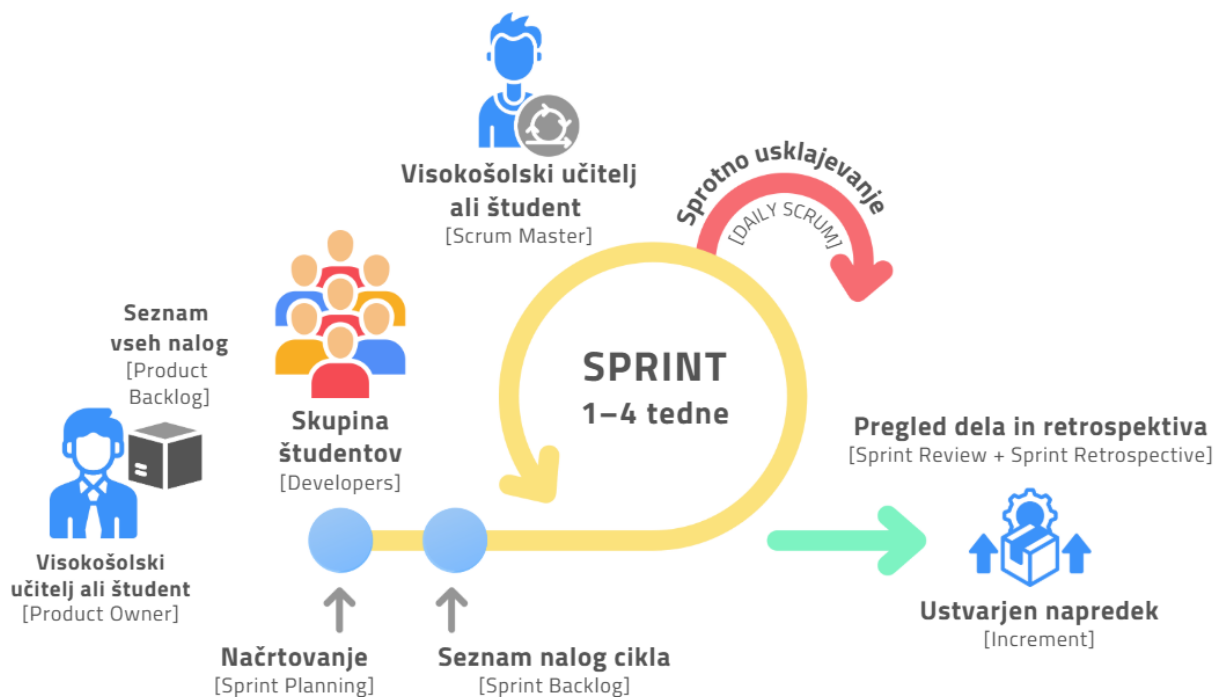
Scrum se dobro dopolnjuje s pristopi **projektnega** in **problemskega učenja**, kjer študenti rešujejo kompleksne naloge **v skupinah**. Več o projektnem in problemskem pristopu je na voljo v strokovnih podlagah v iskalniku gradiv na spletni strani didakt.um.si > Gradiva > [Iskalnik gradiv](#) > [Pedagoški pristopi](#).

Pri tovrstnih pristopih študenti:

- sodelujejo v skupinah ↔ *Developers*
- pripravijo seznam nalog ↔ *Product Backlog*
- obsežno delo razdelijo na več manjših korakov ↔ *Sprinti*
- nadgrajujejo in izboljšujejo izdelek, postopen napredek ↔ *Increment*
- izberejo naloge za posamezni korak ↔ *Sprint Backlog*
- zasledujejo končni cilj, ustvariti nek izdelek ↔ *Product Goal*
- se pogosto usklajujejo, poročajo o delu, izmenjujejo informacije ↔ *Daily Scrum*
- na več časovnih mejnikih predstavijo rezultate in pridobijo povratne informacije (na primer od izvajalca) ↔ *Sprint Review*
- izvedejo refleksijo o procesu dela in prilagajajo delo ↔ *Sprint Retrospective*
- sodelujejo z visokošolskim učiteljem, ki usmerja proces dela in svetuje glede organizacije ↔ *Scrum Master*
- sodelujejo z visokošolskim učiteljem ali imajo vodjo skupine, ki oblikuje vizijo projekta in skrbi za preglednost in usklajenost seznama zastavljenih nalog ↔ *Product Owner*

Posamezni elementi iz ogrodja Scrum lahko pomembno prispevajo k bolj strukturiranemu projektному delu študentov in spodbujajo aktivno sodelovanje v pedagoškem procesu. Pri tem ni nujno uporabljati celotnega ogrodja Scrum, temveč lahko izvajalci izberejo le tiste elemente, ki najbolje podpirajo cilje posamezne učne enote.

Kako lahko elemente pristopa Scrum preslikamo v vloge in delo pri pedagoškem procesu, prikazuje slika 2.



Slika 2: Uporaba ogrodja Scrum v pedagoškem procesu

Možni izzivi in didaktična priporočila:

Pri uvajanju Scrum v pedagoški proces se lahko pojavijo določeni izzivi (Anufrieva, 2020; Gannar in Kilani, 2025). V nadaljevanju izpostavljamo nekatere možne s predlaganimi ukrepi in priporočili.

- **Nejasno razumevanje Scrum in njegovih elementov:** Študenti lahko doživljajo pristop kot zahteven, zlasti zaradi kombinacije vsebinskega učenja in posebnosti pri organizaciji dela. Brez pravega razumevanja namena različnih vlog, dogodkov in artefaktov, lahko aktivnost vodi v površinsko uporabo.

Priporočilo: Na začetku namenite čas strukturirani predstavitvi Scrum (po možnosti na konkretnem primeru in z vizualizacijo strukture procesa dela). Študenti se tako še pred uporabo seznanijo z vsemi elementi in procesom dela. Nekateri avtorji (Gama in Oliveira, 2022; Masson idr., 2023) predlagajo tudi uporabo simulacij in iger (npr. [LEGO4Scrum](#)) za uvodno spoznavanje Scrum, ki lahko povečajo motivacijo in razumevanje pristopa. Spoznavanje agilnih metod tekom študija je za študente koristno, saj preko tega poleg strokovnega znanja razvijajo številne pomembne prenosljive kompetence. Pristop zato uvajajte postopno in prilagodite kompleksnost nalog glede na izkušnje študentov. Kot izvajalec ohranite mentorsko vlogo oziroma vlogo usmerjevalca, ki skupino študentov spodbuja k samoupravljanju in sodelovanju.

- **Težave pri samostojni organizaciji dela:** Študenti lahko imajo težave s samostojnim načrtovanjem, razdelitvijo nalog in z upravljanjem časa, še posebej v začetni fazi, ko se s Scrum prvič srečajo.

Priporočilo: V začetku ponudite več podpore (npr. zgled drugega »backloga«, sodelovanje pri razdelitvi nalog), nato pa postopno spodbujajte skupine k večji avtonomiji. Kot izvajalec ohranite mentorsko vlogo oziroma vlogo usmerjevalca, ki skupino študentov aktivno spodbuja k samoupravljanju in sodelovanju.

- **Neenakomerno sodelovanje v skupini:** Eden najpogosteje izpostavljenih izzivov skupinskega sodelovanja je neenakomeren prispevek članov in težave z neaktivnimi člani. Študenti pogosto izpostavljajo težave v skupinski dinamiki (neučinkovita komunikacija, konflikti).

Priporočilo: Pri oblikovanju skupin oblikujte manjše projektne skupine (3–5 študentov), saj bo že to študentom lajšalo organizacijo dela. Večkrat sodelujte pri pregledovanju skupne table in prispevka posameznikov ter redno preverjajte napredek. Po potrebi aktivno posredujte pri razporejanju nalog. Spremljajte tudi skupinsko dinamiko in ustvarjajte varno okolje, v katerem vam študenti predstavijo izzive pri sodelovanju. Morebitne konflikte lahko predstavite kot priložnost za učenje, kjer študenti ob vaši podpori le-te učinkovito razrešijo.

- **Prevelik ali neustrezen obseg dela v posameznem sprintu:** Študenti lahko precenijo svoje zmožnosti in si zastavijo prevelik obseg nalog za sprint, ali pa si ne znajo razdeliti nalog na manjše enote.

Priporočilo: Začnite s kratkimi sprinti v trajanju enega ali dveh tednov, kar jim bo omogočilo pogostejše preverjanje napredka in sprotno prilagajanje dela. Študente aktivno usmerjajte k razdelitvi nalog na manjše, obvladljive korake.

- **Šibka ali prepozna povratna informacija:** Brez povratne informacije ali s premalo kakovostno povratno informacijo študenti težko izboljšujejo svoje delo. Izvedba pristopa Scrum je lahko zahtevna tudi za izvajalca, ki mora v okviru povratnih informacij nasloviti tako vsebinske kot organizacijske dele.

Priporočilo: Po vsakem sprintu zagotovite usmerjene povratne informacije o zaznanem napredku, saj to študentom pomaga izboljševati rešitve in načrtovati naslednje korake.

- **Površna ali izpuščena refleksija (retrospektiva):** Študenti pogosto podcenijo pomen refleksije ali jo izvajajo zgolj formalno.

Priporočilo: Spodbujajte redno skupinsko in tudi individualno refleksijo o poteku dela. V podporo izvedbi jo lahko strukturirate z usmeritvenimi vprašanji (npr. *Kaj je v skupini delovalo dobro? Katere ovire so se pojavile in jih razrešiti? Ali so bile naloge enakomerno porazdeljene med študente? Kaj ste se naučili o vseh in kaj o načinu sodelovanja? Kako še izboljšati organizacijo dela v naslednjem sprintu?*). Vključite se v kratko obvezno retrospektivo po posameznem sprintu, saj ta študentom pomaga razvijati sposobnost kritičnega razmisleka o lastnem načinu dela in možnostih napredka. Študenti naj o glavnih ugotovitvah poročajo, pri tem pa kot izvajalec spodbujajte varno okolje za izražanje mnenj in predlaganje rešitev.

6.1 eduScrum kot pedagoška prilagoditev Scrum

Za pedagoški prostor je zanimiv tudi eduScrum¹, ki predstavlja prilagoditev Scrum ogrodja za uporabo v izobraževanju. Pri tem ohranja osnovno strukturo Scrum, vendar jo prilagaja pedagoškemu kontekstu. Po vzoru Scrum vodiča je bil razvit tudi [eduScrum vodič](#), ki je dosegljiv na uradni spletni strani skupnosti eduScrum. Pristop je namenjen vodenju učečih se tako, da se odgovornost za učenje v večji meri prenaša z učitelja na študente, pri čemer je poudarek na učinkovitem doseganju pričakovanih učnih izidov. V ospredju ostajajo temeljna načela empirizma, značilna za Scrum: transparentnost, sprotno preverjanje napredka in prilagajanje dela.

Za razliko od izvirnega Scrum, ki je usmerjen v razvoj kompleksnih produktov, je eduScrum usmerjen v **učenje, sodelovanje in osebni razvoj študentov**. Ob večji odgovornosti študentov za študijsko delo se krepijo motivacija, sodelovanje in kakovost učenja. Avtorji eduScrum izpostavljajo koristno povezavo s **personaliziranim učenjem** in razvojem štirih ključnih kompetenc 21. stoletja: **ustvarjalnosti, sodelovanja, komunikacije in kritičnega mišljenja**.

Če Scrum strukturira delo, eduScrum strukturira učenje. V eduScrum so prisotne t. i. »*Celebration Ceremonies*«, ki predstavljajo ključne dogodke v procesu učenja. Ti so primerljivi z dogodki v Scrum (načrtovanje, delo v krajših ciklih, pregled in refleksija), le da so prilagojeni pedagoškemu kontekstu. Poudarjajo motivacijo, sodelovanje ter prepoznavanje dosežkov. Načrtovanje je usmerjeno v to, da študent razume pričakovane učne izide, študijsko delo se osredotoča na njihovo doseganje, kratki sprotne sestanki pa so usmerjeni v to, da študenti razvijajo odgovornost do učenja. Pregledi omogočajo, da študenti pridobijo povratno informacijo, retrospektiva pa krepi refleksijo.

V eduScrum je nekoliko drugačna tudi razporeditev vlog. Učitelj je nosilec **hibridne vloge**, ki združuje odgovornosti vlog Product Owner in Scrum Master. Je tisti, ki izbira vsebine in opredeli učne izide, določa kriterije uspeha in preverja njihovo doseganje. Ob tem študentom predstavi eduScrum in nato spremlja predvideno strukturo učenja. Pri tem je še vedno odgovoren za podporo vsakemu študentu (npr. preko dodatne razlage, podajanja primerov) in za spodbujanje sodelovanja znotraj skupine ter med skupinami. V tem kontekstu lahko kot »produkt« razumemo študentovo znanje, napredek po vsakem ciklu učenja pa njegovo nadgradnjo. Učitelj pri tem postavi okvir, znotraj katerega študenti samostojno načrtujejo delo. Kriteriji uspeha »*Celebration Criteria*« predstavljajo jasne kriterije uspeha in so primerljivi z rubrikami. Skozi jasno zastavljene kriterije uspeha eduScrum omogoča večjo preglednost **pričakovanj**, podpirajo **samoregulacijo** in **odgovornost študentov** za študijske rezultate (učne izide).

Študentske skupine so tudi v eduScrum **samoupravljajoče**. Priporočena velikost skupine je od štiri do pet študentov, oblikuje pa jih učitelj, ki predhodno preuči njihove značilnosti. Znotraj skupine ima eden od članov posebno vlogo, t. i. »*Team Captain*«. To ni vodja skupine, temveč

¹eduScrum® je registrirana znamka ustanovitelja skupnosti eduScrum team, Willyja Wijnandsa.

enakovreden član, ki ima zadolžitev, da skrbi za preglednost dela z rednim beleženjem vseh nalog v skupen seznam potrebnih nalog za doseganje učnega cilja na posebno tablo. Učitelj je tisti, ki opolnomoči izbranega študenta za to vlogo. Del odgovornosti za usmerjanje učenja ostaja na učitelju, se pa s časom in izkušnjami lahko več odgovornosti prenese na tega študenta.

Posebnost eduScrum je, da poleg osnovnih elementov Scrum poudarja tudi nekatere dodatne didaktične vidike oz. priložnosti:

- a) **Načrtno oblikovanje skupin («*Team Formation*«)**: Učitelj premišljeno oblikuje heterogene skupine študentov glede na različne potrebe in tudi različne interese, tako da se člani med seboj ustrezno dopolnjujejo. V pedagoškem smislu to omogoča učenje drug od drugega, razvoj medosebnih kompetenc in večjo vključenost.
- b) **Vizualna tabla («*the Flap*«)**: Flap je skupen pregleden seznam vseh nalog, ki so potrebne za doseg učnega izida. Običajno vključuje kronološko urejene posamezne naloge in stanje nalog, dogovore, kriterije uspeha in napredek skupine. Za ažurnost skrbi »Team Captain«, vendar so za vsebino odgovorni vsi člani skupine, vpogled pa ima tudi učitelj. Ta tabla se tekom študijskega dela dopolnjuje oz. nadgrajuje. Preko vizualizacije lažja vpogled v napredek skupine.
- c) **Načrtovanje dela na podlagi učnih izidov**: V eduScrum je delo (učenje) v posameznih sprintih jasno povezano z vnaprej določenimi učnimi izidi. Namesto da bi se študenti osredotočali zgolj na izvedbo nalog, je v ospredju vprašanje, kaj naj bi se z učenjem doseglo. Tak pristop spodbuja tudi razmislek o smislu nalog in lažji povezavi z učnimi izidi.
- d) **Spremljanje napredka («*Run Up Chart*«)**: »Run Up Chart« je grafični prikaz za spremljanje napredka skupine skozi čas, ki kaže, koliko dela je skupina že opravila glede na razpoložljiv čas (npr. število srečanj oziroma kontaktnih ur). Je del table »Flap«. Študenti lahko na podlagi grafa sami ocenijo, ali napredujejo skladno z načrtom in po potrebi prilagodijo svoje delo.
- e) **Kratko usklajevalno srečanje («*Stand up*«)**: Stand up je kratko (približno 5 minut) srečanje skupine, na katerem člani skupine uskladijo delo in prepoznajo morebitne ovire. V podporo so usmeritvena vprašanja: Kaj sem naredil/-a od zadnjega srečanja? Kaj bom naredil/-a danes? Katere ovire motijo moje ali skupinsko delo? V pedagoškem smislu ta element spodbuja redno komunikacijo in sodelovanje ter razvija odgovornost posameznika do skupine. Stand up ima v eduScrum podobno vlogo kot Daily Scrum v izvornem Scrum, vendar običajno ne poteka vsak dan, saj se skupina ne srečuje tako pogosto.

- f) **Dogovori članov skupine (»Definition of Doing«):** Gre za dogovore študentov, članov skupine o tem, kdaj je njihovo delo res zaključeno. Povezuje se z »Definition of Done«. Ti dogovori se povezujejo z opredeljenimi kriteriji uspeha. Člani skupine si na primer zastavijo dogovore: »vsak član skupine je sposoben samostojno povzeti vsebinsko poglavje«. Ta element povečuje jasnost pričakovanj med študenti in zmanjšuje nesporazume.
- g) **Dogovori o spodbudnem okolju (»Definition of Fun«):** Predstavlja dogovore o tem, kakšno okolje si skupina želi ustvariti, pri čemer je v ospredju pozitivno vzdušje, v katerem je prisotna na primer odprta komunikacija, spoštovanje in medsebojna podpora. Didaktično je ta element posebej pomemben, saj spodbuja psihološko varnost in krepi sodelovanje in zaupanje. Gre za element, ki presega zgolj organizacijo dela in vključuje tudi socialno-čustveni vidik učenja, kar je pri skupinskem delu ključnega pomena.

7 Primeri uporabe v izobraževanju

V literaturi najdemo več primerov uporabe Scrum in njegove prilagoditve eduScrum na različnih študijskih področjih. Spodaj so predstavljeni izbrani primeri iz visokošolskega prostora.

Fizika:

V študiji na področju poučevanja fizike ([Lourakis in Petridis, 2023](#)) so Scrum uporabili za organizacijo skupinskega dela študentov pri reševanju nalog (Physics II) v času študija na daljavo. Študenti so delali v skupinah, uporabljali kratke iteracije in redno preverjali napredek. Del študentov pa ni sodeloval v Scrum načinu dela. Rezultati so pokazali večjo vključenost študentov v proces učenja, izboljšanje mehkih veščin (komunikacija, timsko delo, reševanje problemov) in pozitiven odnos študentov do sodelovalnega dela. Študenti, ki so sodelovali v Scrum, so imeli tudi višje ocene. Avtorja opisujeta zaznano postopno izgradnjo varnosti znotraj skupine. Študenti so bili sprva bolj zadržani, preko spodbud in rednih srečanj pa so se sčasoma opogumili ter odprto izrazili kje zaznavajo, da imajo šibke točke. Avtorja sta zaznala tudi izboljšanje veščin v organizaciji in upravljanju s časom.

Biologija:

Naslednji primer se povezuje s študijskim predmetom biologije na univerzitetnem izobraževanju (North Dakota State University), kjer so študenti opravljali predmet s t. i. CURE formatom (angl. *course-based undergraduate research experience*). V okviru predmeta so študenti v skupinah delali na raziskovalnih projektih, pri čemer so uvedli prilagojeno različico Scrum. Visokošolski učitelj je prevzel vlogo Product Owner in je določal cilje ter pričakovanja. Asistent je prevzel vlogo Scrum Master, študente podrobno seznanil s Scrum okvirjem in kasneje skrbel za proces dela. Študenti so bili v skupine razporejeni na podlagi vsebinskega interesa. Pri delu so samostojno načrtovali tedenske naloge. Vsaka skupina je vodila posebne zapiske (v skupni rabi preko Google dokumentov), kjer so beležili, kaj kdo dela, kaj je opravljeno. V okviru kontaktnih ur so namenili 15 minut Scrum aktivnostim (Sprint Retrospective in Sprint Planning na začetku, ter kratek sestanek ob koncu). Izvajalci, avtorji raziskave, so ugotovili, da so ob takem načinu dela študenti poročali o boljši komunikaciji, boljši organizaciji, čutili so tudi več odgovornosti do dela. Način organizacije dela s Scrum ogrodjem bi študenti uporabili tudi v prihodnje (tudi kasneje v delovnem okolju). Med izpostavljenimi pomanjkljivostmi s strani manjšega deleža študentov je bilo zaznati, da so nekateri zvezek z zapiski o procesu dela doživljali kot nepotrebno birokratsko delo. Raziskava o predstavljenem primeru ([Hackerson, Evenson in Momsen, 2025](#)) je bila sicer izvedena na zelo majhnem vzorcu (17), zato rezultatov ne gre posploševati.

Biologija (primer iz raziskovalnega okolja):

Primer prihaja iz [laboratorija za biologijo na Leiden University](#), kjer je interdisciplinarna skupina raziskovalcev sodelovala pri kompleksnem raziskovalnem projektu s področja biologije. V raziskovalni skupini so delo organizirali po principih Scrum z namenom izboljšanja sodelovanja, preglednosti dela in usklajevanja različnih strokovnjakov. Delo je potekalo v sprintih (4–6 tednov), pred začetkom posameznega sprinta pa so raziskovalci skupaj določili naloge in cilje. Redna kratka srečanja so bila namenjena usklajevanju napredka, prepoznavanju težav in načrtovanju nadaljnjega dela. Pomembno vlogo je imela tudi vizualna tabla za spremljanje nalog in napredka projekta. Avtorji poročajo, da je uporaba Scrum izboljšala sodelovanje med člani raziskovalne skupine, povečala preglednost dela ter olajšala koordinacijo kompleksnega interdisciplinarnega raziskovanja. Raziskovalci so posebej izpostavili, da je Scrum zaradi iterativnega načina dela primeren tudi za raziskovalno delo, kjer so cilji pogosto nejasni ali se med procesom spreminjajo. Projekt so uspešno zaključili v razmeroma kratkem času (10 mesecev od začetka raziskave do sprejetja članka v objavo). Primer kaže, da je Scrum uporaben tudi v akademskem raziskovalnem okolju, zlasti pri projektih, kjer sodelujejo interdisciplinarne skupine.

Ekonomske in poslovne študije:

V magistrskem programu s področja digitalne transformacije so eduScrum uporabili pri predmetu agilnega projektne managementa, ki ga je obiskovalo 25 študentov ([Neumann in Baumann, 2021](#)). Študenti so v okviru predmeta reševali realne probleme gospodarstva. V sodelovanju s štirimi podjetji so izvajalci pripravili projektne naloge, ki so bile oblikovane s konkretnimi izzivi (npr. analiza pojava, razvoj modela za uvedbo novega pristopa v podjetju). Naloge so bile strukturirane v obliki »backloga«, kar je študentom omogočilo pregledno načrtovanje dela. Študenti so bili nato razdeljeni v štiri skupine (eduScrum ekipe), vsaka skupina je delala na enem projektu. Delo je potekalo iterativno, v krajših ciklih, pri čemer so študenti samostojno načrtovali naloge, spremljali napredek in sproti prilagajali delo. Pristop so izvajali najprej v živo, nato pa tudi na daljavo (v času zaprtja). Avtorja poročata, da so študenti pozitivno vrednotili delo na realnih problemih iz prakse (relevantnost učenja), v članku opisujeta tudi visoko stopnjo angažiranosti in aktivne vključenosti ter o primernosti uporabe pristopa za izvedbo pedagoškega procesa na daljavo, ki se je izkazala kot ustrezna.

Umetnost:

Primer s področja študija likovne pedagogike ([Leyva Haza, Hurtado Guerra in Véliz, 2022](#)) ponuja vpogled v vpeljavo eduScrum pri predmetu umetniškega ustvarjanja, kjer so študenti sodelovali pri razvoju skupnega umetniškega projekta. Cilj je bil preseči tradicionalno individualno delo in spodbuditi sodelovalno ustvarjanje. Projekt je bil usmerjen v ustvarjanje tridimenzionalnega umetniškega izdelka, pri čemer so študenti sodelovali pri načrtovanju,

izvedbi in sprotne vrednotenju dela. Proces je vključeval jasno razdeljene naloge, kriterije uspeha (v obliki rubrik) in redno spremljanje napredka. Poseben poudarek je bil na sodelovanju, izmenjavi idej ter skupnem odločanju v ustvarjalnem procesu. Rezultati so pokazali postopno izboljševanje kakovosti izdelkov skozi posamezne faze projekta ter pozitiven vpliv na motivacijo in sodelovalno učenje. Študenti so razvijali tako umetniške kot tudi socialne in organizacijske kompetence. Primer kaže, da je eduScrum primeren tudi za področja, kjer prevladuje individualno delo, saj omogoča strukturirano sodelovanje in podpira ustvarjalni proces skozi iterativno delo in sprotne refleksije.

Turizem:

Primer uporabe Scrum s študenti turizma v okviru poučevanja strokovnega angleškega jezika, kjer so študenti razvijali komunikacijske kompetence skozi projektno delo. Študenti so sodelovali pri reševanju nalog, povezanih z realnimi situacijami v turizmu (npr. priprava turističnih produktov, predstavitev turističnih destinacij, komunikacija s strankami). Učenje je potekalo iterativno, v krajših ciklih, z rednim preverjanjem napredka in prilagajanjem dela. Posebnost pristopa je bila integracija vsebinskega in jezikovnega učenja, pri čemer so študenti razvijali strokovno terminologijo, komunikacijske spretnosti in sposobnost sodelovanja v timu. Rezultati so pokazali, da tak način dela spodbuja večjo aktivno vključenost študentov, izboljšuje komunikacijo v tujem jeziku ter povečuje občutek relevantnosti učenja, saj so naloge povezane z realnimi delovnimi situacijami. Avtorja ([Gragera in Soto Déniz, 2024](#)) poudarjata zaznane bolj dinamično, sodelovalno in praktično usmerjeno učenje, ki študente bolje pripravlja na zahteve delovnega okolja v turizmu. Študenti so skozi anketo poročali o lastnih zaznavah, med drugim o izboljšanih kompetencah, izboljšanih veščinah organizacije dela, timskega dela, upravljanja s časom idr. Zaradi izvedbe, ki je temeljila na projektnem delu, se je po poročanju avtorjev zmanjšalo tudi izostajanje.

Računalništvo:

V okviru magistrskega študija računalništva (predmet podatkovnega rudarjenja) so eduScrum uporabili za organizacijo učnega procesa z namenom povečanja aktivne vključenosti študentov ([Andersson in Kroisandt, 2025](#)). Študenti so delali v manjših skupinah, organiziranih kot eduScrum ekipe, pri čemer so prevzeli odgovornost za načrtovanje in izvedbo dela. Učni proces je bil strukturiran v sprinte, ki so vključevali uvodni »Kick-off«, samostojno delo v skupini (»Doing«), pregled rezultatov (»Review«) in refleksijo (»Retrospective«). Naloge so bile organizirane v obliki backloga, izvajalec pa je imel vlogo »Product Owner«, ki je določal učne cilje in kriterije kakovosti, ter ohranil vlogo mentorja, ki je skupine spremljal in podpiral. Posebnost izvedbe je bila, da je delo potekalo samostojno (izven kontaktnih ur), kar je dodatno poudarilo odgovornost študentov za lastno učenje. Izvedli so 11 sprintov v trajanju enega tedna. Vsakemu sprintu je sledil pregled napredka, v okviru katerega je izvajalec izvedel kviz, kjer so posamezne skupine tekmoval med seboj.

Zadnja dva tedna sta bila namenjena razreševanju odprtih vprašanj študentov. Rezultati so pokazali večjo aktivno vključenost študentov v proces učenja, pozitivno zaznavanje sodelovalnega načina dela in potencialno zmanjšanje neuspešnosti (višja prehodnost v primerjavi s tradicionalno izvedenim predmetom). Avtorja sta opozorila na zaznано zmanjšanje aktivnosti pri študentih skozi čas, za kar sta predlagala vključitev več obveznih nalog, ki bi prispevali k vzdrževanju motivacije skozi celotno obdobje izvedbe.

8 Viri

Andersson, C. in Kroisandt, G. (2025). A First Analysis of Students' Feedback to an eduScrum-Based Course in a Computer Science Master's Program. V *2025 MIPRO 48th ICT and Electronics Convention*. IEEE. <https://doi.org/10.1109/MIPRO65660.2025.11131929>

Čížek, T. in Vojír, K. (2023). Strengths and weaknesses of using the eduScrum method in science education. V Koperová, D. in Rusek, M. (ur.), *Project-based education and other activating strategies and issues in science education XXI, 1*, 45–53. Charles University, Faculty of Education. https://pages.pedf.cuni.cz/pbe/files/2024/09/PBE2023_final.pdf

Fekonja, A. (2023). *Primerjava Kanban in Scrum* [Diplomsko delo, Univerza v Mariboru, Fakulteta za elektrotehniko, računalništvo in informatiko UM]. Digitalna knjižnica Univerze v Mariboru. <https://dk.um.si/lzpisGradiva.php?id=86341>

Gannar, S. in Kilani, C. (2025). Difficulties revealed in the teaching-learning process of scrum. *Social Sciences & Humanities Open*, 11, 101357. <https://doi.org/10.1016/j.ssaho.2025.101357>

Gama, K. in Oliveira, H. (2022). An Experience Report on Teaching Scrum Principles in a Playful Way through Distant Collaboration with Online Whiteboards. V *Anais do XXXVI Simpósio Brasileiro de Engenharia de Software*, 143–152. Porto Alegre: SBC.

Gragera, R. in Soto Déniz, I. (2024). Proposing the Integration of Scrum and ESP for EFL Tourism Students: Theoretical Framework and Methodological Approach. *Language Value*, 17(2), 24–59. <https://doi.org/10.6035/languagev.8474>

Hackerson, E. L., Evenson, A., in Momsen, J. L. (2025). *Better Together: Using Scrum to Enhance the Group Work Experience in a Biology Education CURE*. *CourseSource* 12. <https://doi.org/10.24918/cs.2025.33>

Leyva Haza, J., Hurtado Guerra, R. in Veliz War, Y. (2022). Projects with EduScrum techniques in the training of Art Education teachers. *Mendive-revista de educacion*, 20(4). https://mendive.upr.edu.cu/index.php/MendiveUPR/article/download/3027/html_1?inline=1

Lourakis, E. in Petridis, K. (2023). Applying Scrum in an Online Physics II Undergraduate Course: Effect on Student Progression and Soft Skills Development. *Education Sciences*, 13(2), 126. <https://doi.org/10.3390/educsci13020126>

Masson, E. T. S., Calazans, A. T. S., Bandeira, I. N., Silva, G. R. S. in Canedo, E. D. (2023). Scrum in practice: City reconstruction as a pedagogical game challenge. V E. D. Canedo, D. de Paula Porto, F. L. L. de Mendonça, R. T. de Sousa Júnior, M. P. Barcellos, I. de Sousa Santos, S. S. Reinehr, S. Soares, U. Kulesza, É. F. de Souza, A. Albuquerque, C. I. M. Bezerra, R. P. dos Santos, A. F. Garcia, S. D. Costa in A. G. S. S. Neto (ur.), *Proceedings of the XXII Brazilian*

Symposium on Software Quality (SBQS 2023) (str. 321–331). ACM.

<https://doi.org/10.1145/3629479.3629480>

Mitchell, I. (9. 11. 2015). https://en.wikipedia.org/wiki/File:Scrum_Framework.png

Neumann, M. in Baumann, L. (2021). Agile methods in higher education: Adapting and using eduScrum with real world projects. V *2021 IEEE Frontiers in Education Conference (FIE)*.

<https://arxiv.org/pdf/2106.12166>

Quantitative Single-Cell Biology Lab, Leiden University. (2020). Scrum for science: A framework for collective research. <https://crosstalk.cell.com/blog/scrum-for-science-a-framework-for-collective-research>

Spichkova, M. (2025). Agile and Student-Centred Teaching of Agile/Scrum Concepts. *Procedia Computer Science*, 270, 3558–3567. <https://doi.org/10.1016/j.procs.2025.09.481>

Schwaber, K. in Sutherland, J. (2020). *The Scrum Guide*. <https://scrumguides.org/>

Wijnands, W. idr. (2020). *The eduScrum guide*. <https://eduscrum.org/>