Razvoj informacijskih sistemov

Informatika in podatkovne tehnologije

UNI, 2. letnik

Študijsko leto 2025 / 2026

Luka Pavlič, UM, FERI

Organizacije ekipnega razvoja
17. in 24. 10. 2025

Dogovori znotraj razvojnih ekip

Okolje za upravljanje sprememb

Dokumentacija za razvijalce

Dokumentacija za operativo

Preden se „zaletimo“ v kodo…?

Priprava okolja (strojna oprema, programska oprema, dokumentacija)

Nastavitev orodij

→ Implementacija

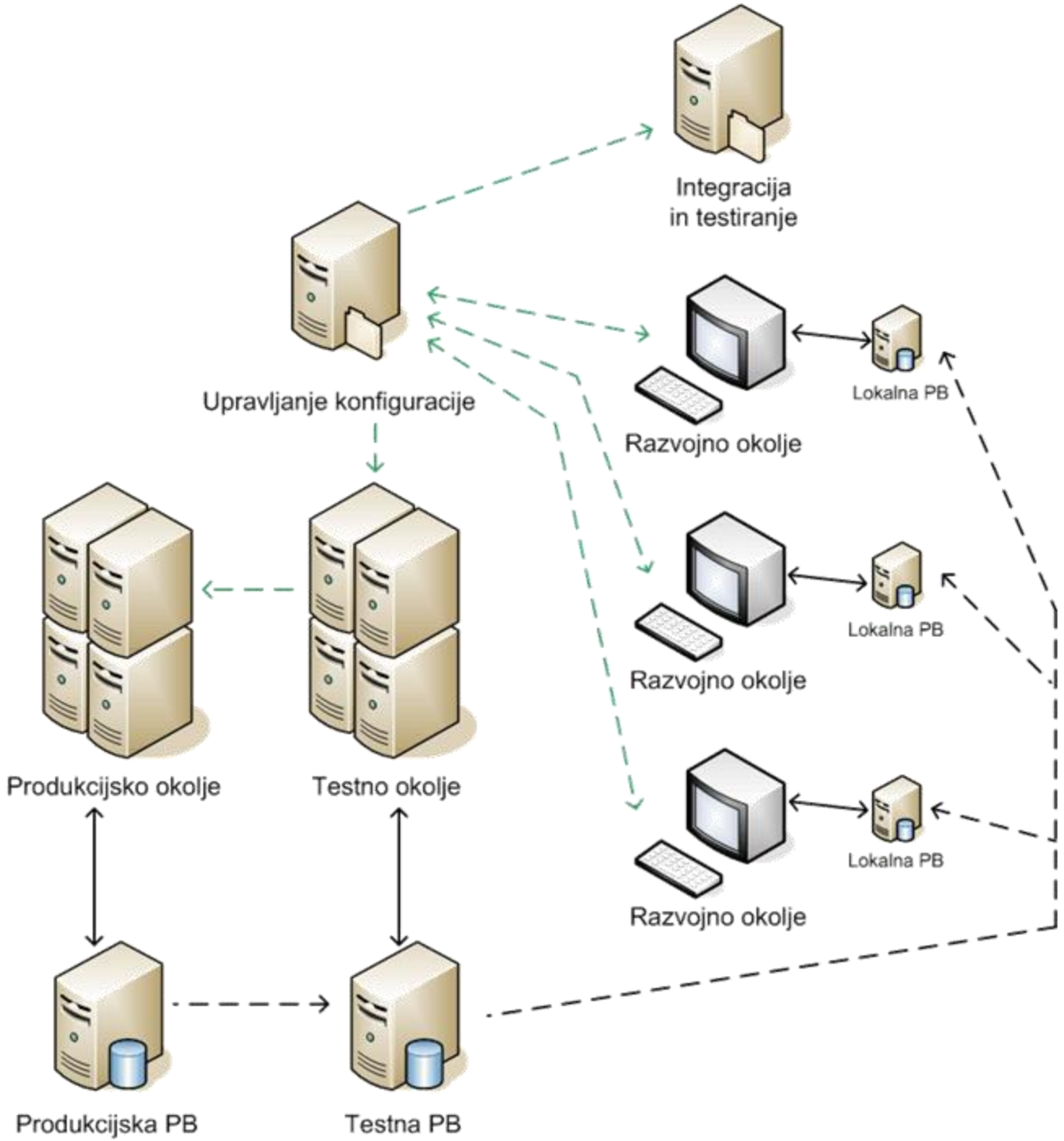
Med tem:

- Razvoj testnih primerkov
- Poganjanje testnih primerkov
- Integracija rešitve
- Priprava paketov (build-ov)

Razvoj informacijskih sistemov je skupinski trud!

Organizacija razvojnega okolja

Se še vrnemo sem…



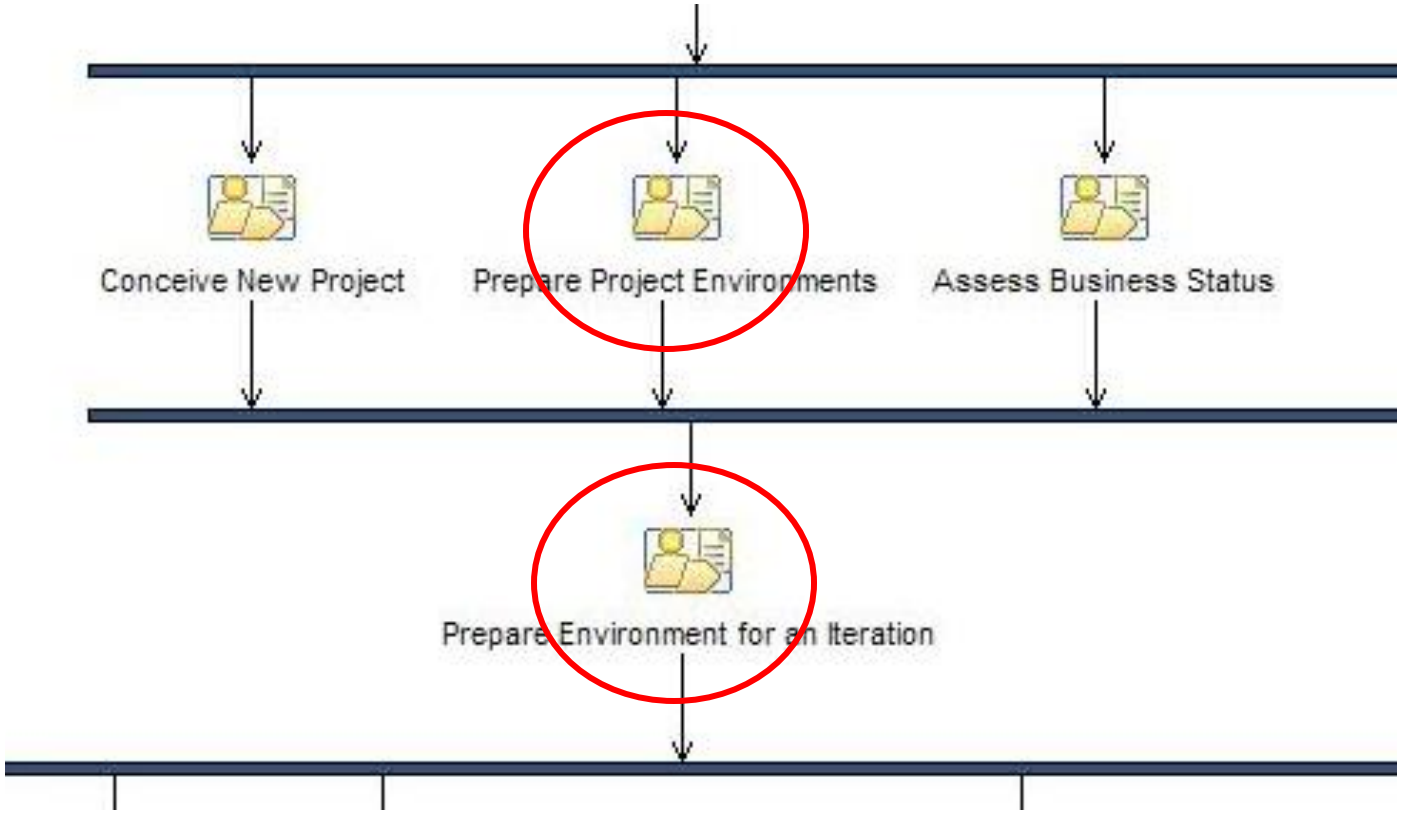
Samo eden izmed uveljavljenih modelov.

Upravljanje konfiguracije (to že znate…)

Vnaprejšnja priprava

- Skupnega repozitorija
- Predlog dokumentov,
- Imeniške strukture,
- Pravil “igre”
- …

<u>Večino tega opravimo že na začetku projekta!</u>



Sistemi za upravljanje sprememb

Cilji:

- Skupen repozitorij datotek
- Propagacija sprememb
- Upravljanje konfliktov

Pristopi:

- Klasični zakleni-in-urejaj
- Prenesi-uredi-objavi
- Veje / reševanje sprememb

Git
Subversion (SVN)
Mercurial
Perforce Helix Core
TFS (Team Foundation Server)
CVS (Concurrent Versions System)
Bazaar
ClearCase
SourceSafe
…

Odličitve pri kreiranju repozitorija

T.i. “monorepo”?
- Več projektov v enem repozitoriju?

Način razvoja?
- T.i. Feature-based (uporaba vej in kasnejše zlivanje)
- T.i. Trunk-based (objava neposredno v glavno vejo in sprotno zlivanje)

Vsebina repozitorija
- Datoteke, ki v skupen repozitorij ne spadajo
- Deli datotek, ki v skupen repozitorij ne spodajo

Javni repozitoriji?
- GitHub, GitLab, Bitbucket, SourceForge, Gitea…

Pazite: pogosto imamo ločene repozitorije (in sisteme) za delo na kodi in dokumentaciji

Zelo enostavno iskanje po javnih repozitorijih…

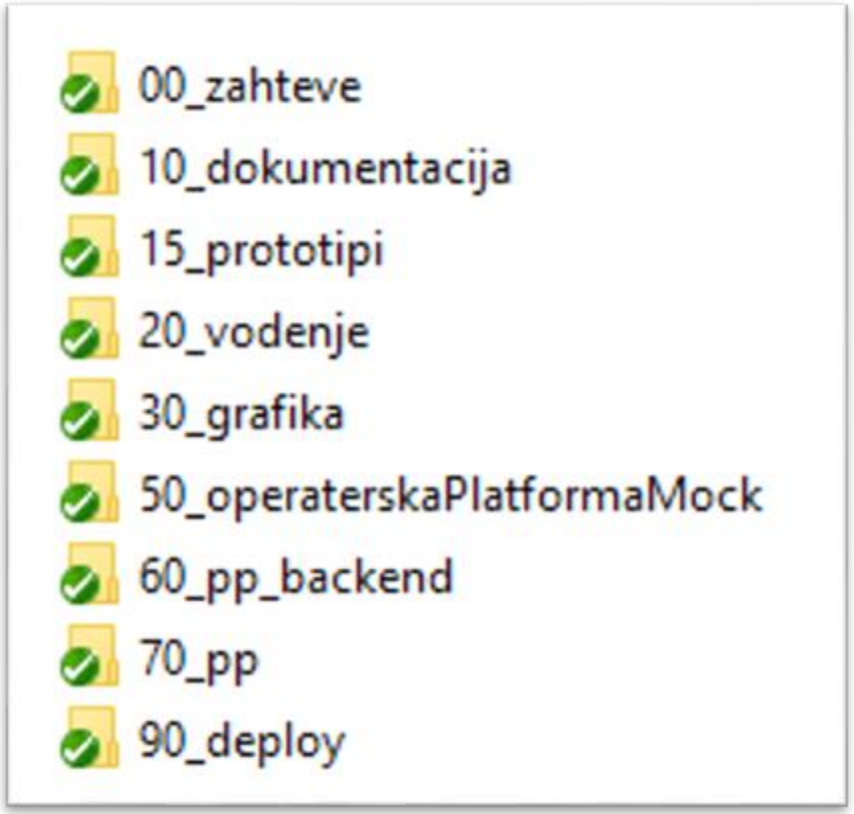
```
22  # PROD
23  prod.quarkus.datasource.reactive.url=postgresql:[redacted]postgres.database.azure.com:543[redacted]sslmode=require
24  prod.quarkus.datasource.username=[redacted]
25  prod.quarkus.datasource.password=[redacted]
```

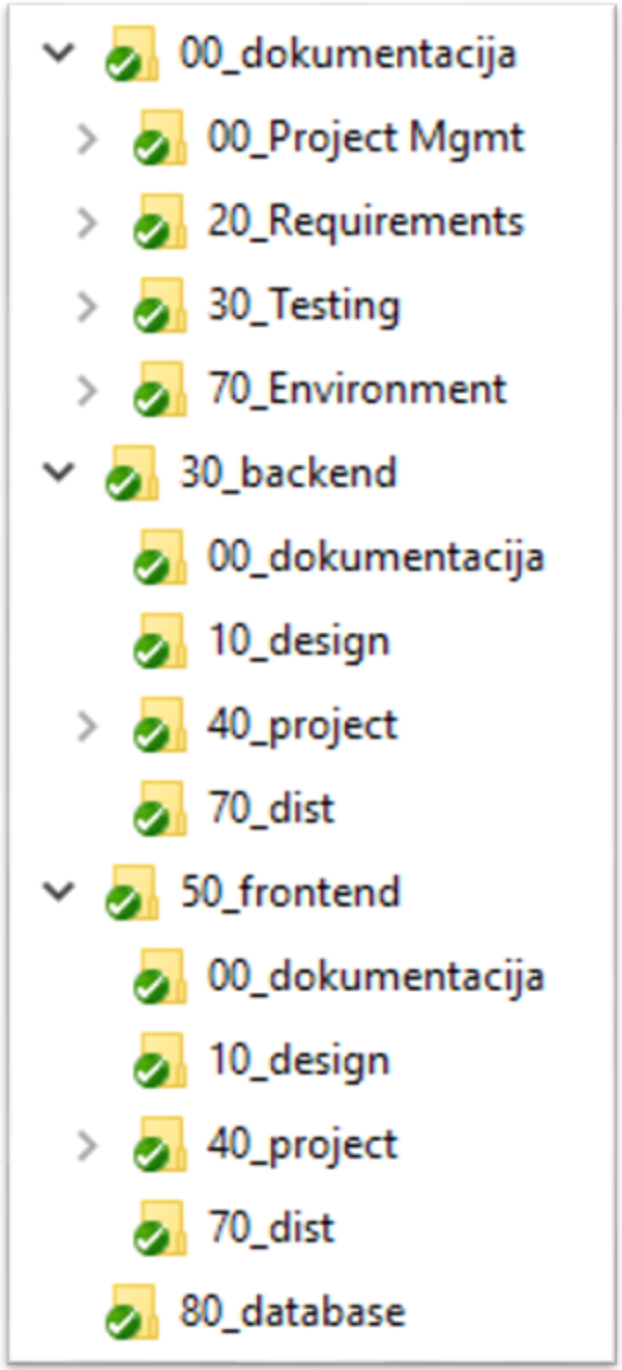
frontend > scm > .env

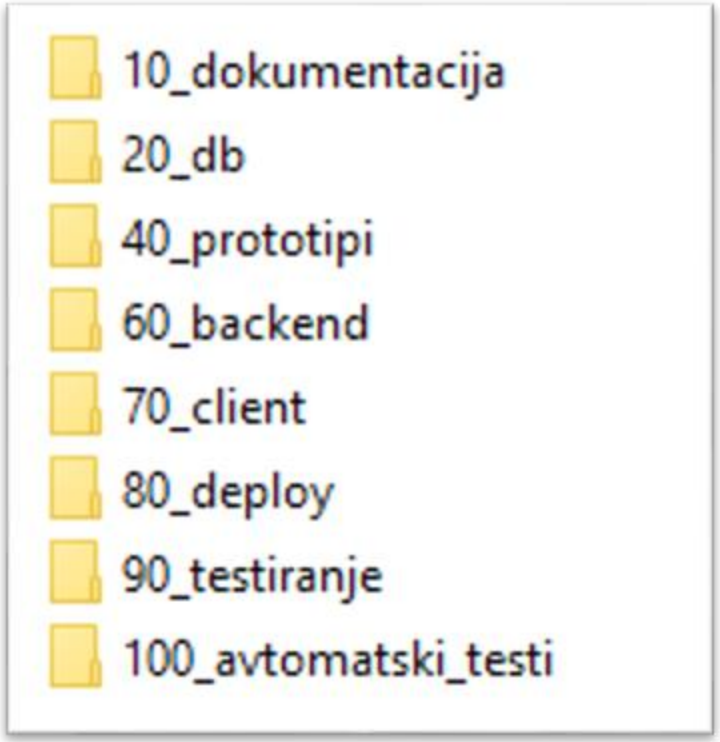
```
1  NEXT_PUBLIC_FIREBASE_API_KEY=[redacted]
2  NEXT_PUBLIC_FIREBASE_AUTH_DOMAIN=[redacted].firebaseapp.com
3  NEXT_PUBLIC_FIREBASE_PROJECT_ID=[redacted]
4  NEXT_PUBLIC_FIREBASE_STORAGE_BUCKET=[redacted].appspot.com
5  NEXT_PUBLIC_FIREBASE_MESSAGING_SENDER_ID=[redacted]
6  NEXT_PUBLIC_FIREBASE_APP_ID=1[redacted]860d
7  NEXT_PUBLIC_API_URL="http://[redacted]:8080"
8
```

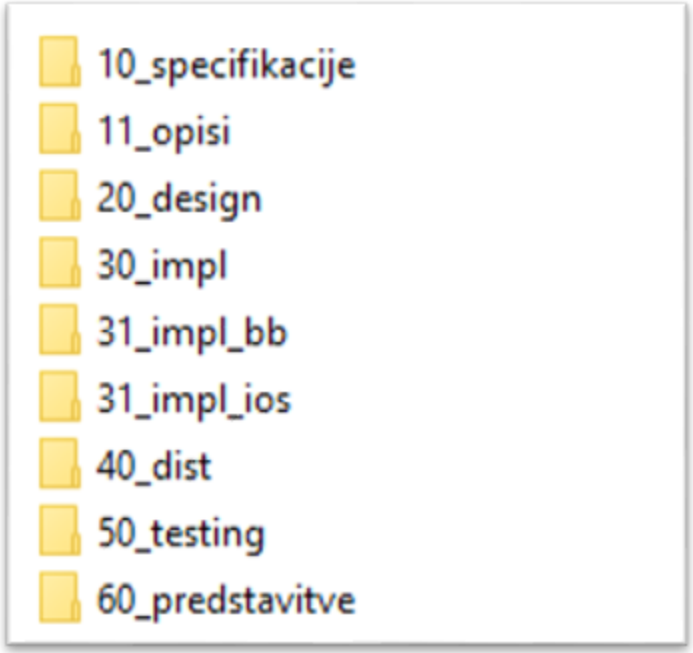
Tudi v privatne (“privatne“) se to NE daje!

Nekaj primerov







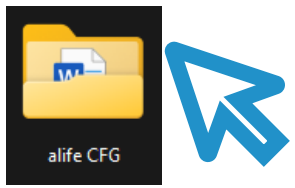


- 00_Dokumentacija
  - 00_Project Management
  - 01_Business Modeling
  - 02_Requirements
  - 03_Analysis and Design
  - 04_Test
  - 05_Deployment
  - 06_Configuration and Change Management
  - 07_Environment
  - 09_Promocija
- 01_Poročila
  - Changelog
  - Google Play
  - Jeziki

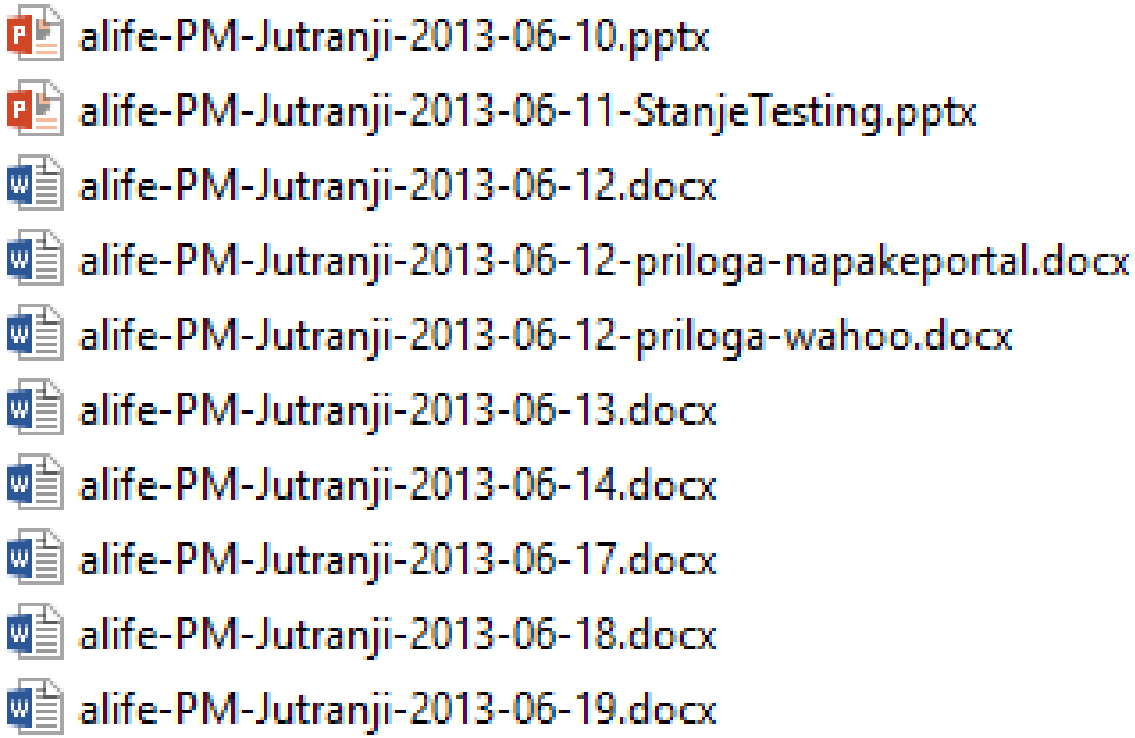
- AndroidV2
  - 10_dokumentacija
  - 20_design
  - 30_project
  - 30_project_alpha
  - 31_android_demo
  - 31_android_full
  - 31_android_lite
  - 32_blackberry_full
  - 32_blackberry_lite
  - 34_library_common_and
  - 34_library_facebook
  - 34_library_google_maps2
  - 34_library_web_maps
  - 35_test
  - 40_dist
  - 60_concepts
  - 99_old

- iOS
- Portal
  - 00_temp
  - 01_dokumentacija
  - 05_design
  - 15_model
  - 20_komponente
  - 30_portal
  - 50_test
- PortalTestRestEndpoint
- Product Backlog
- Tablica
- WindowsPhone8
- Wowza

https://gitlab.com/luka.pavlic/bestmeddev



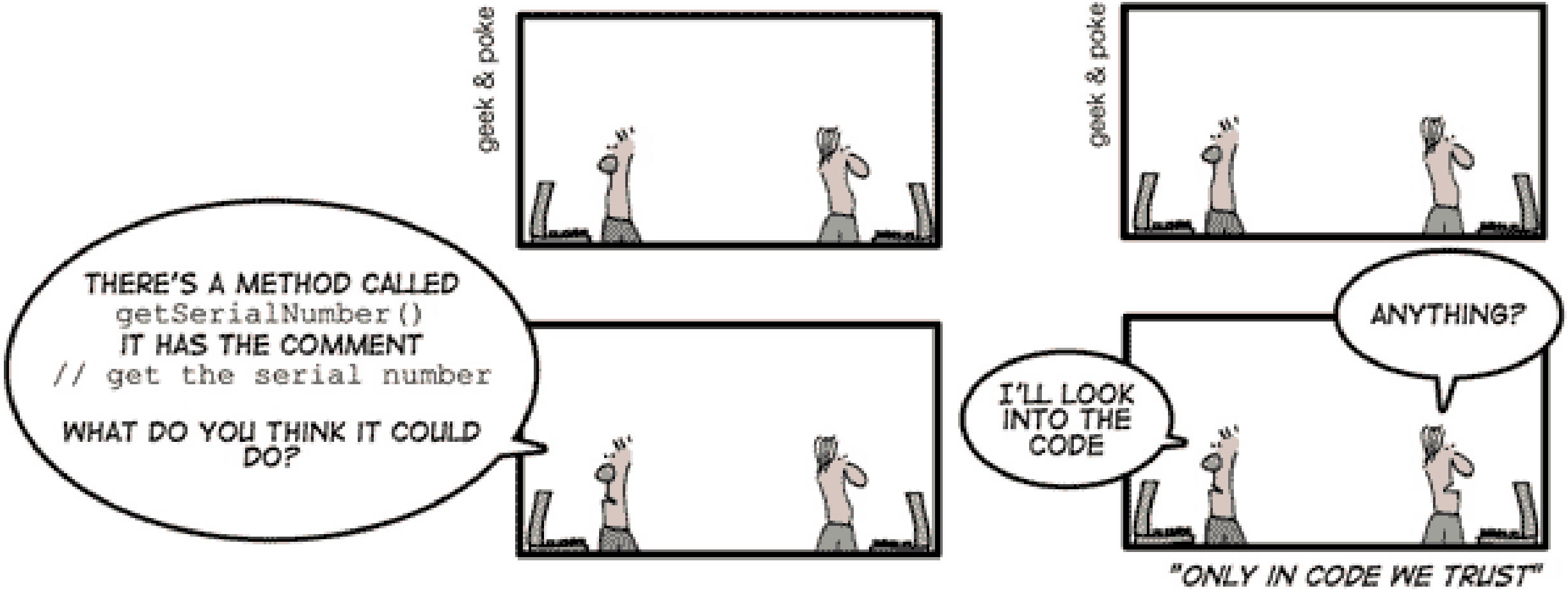
00_Dokumentacija\00_Project Management\Sestanki\Interni\Jutranji sestanki\2013-06



Imeniška struktura, poimenovanja datotek in predloge so dokumentirani!


document.getElementById(div).
else if (i==2)
{
var atpos=inputs[i].indexOf("@");
var dotpos=inputs[i].lastIndexOf(".");
if (atpos<1 || dotpos<atpos+2
document.getElementById('errfmail').
else
document.getElementById(div).
Zdaj pa čisto zares ☺

THE REAL CODER



"ONLY IN CODE WE TRUST"

```java
import java.text.SimpleDateFormat;
import java.util.Calendar;
import java.util.GregorianCalendar;

public class Oseba {

public Oseba() {
    this("", "","");
}

public Oseba(String ime, String priimek, String email) {
    this.email = email;
    this.ime = ime;
    this.priimek = priimek;
    this.spol = "";
    datumRojstva=new GregorianCalendar();
    datumVpisa=new GregorianCalendar();
}
...
```

```java
...
private String ime;

private String priimek;

private String email;

private String spol;

private Calendar datumRojstva;

private Calendar datumVpisa;

public String getIme() {
    return ime;
}

public void setIme(String ime) {
    this.ime = ime;
}
...
```

```
import java.text.SimpleDateFormat;
import java.util.Calendar;
import java.util.GregorianCalendar;

public class Oseba {

public Oseba() {
    this("", "","");
}

public Oseba(String ime, String priimek, String email) {
    this.email = email;
    this.ime = ime;
    this.priimek = priimek;
    this.spol = "";
    datumRojstva=new GregorianCalendar();
    datumVpisa=new GregorianCalendar();
}
...
```

```
...
private String ime=null;

private String priimek=null;

private String email=null;

private String spol=null;

private Calendar datumRojstva;

private Calendar datumVpisa;

public String getIme() {
    return ime;
}

public void setIme(String ime) {
    this.ime = ime;
}

public static final String MOSKI="M";
public static final String ZENSKA="Z";

...
```

```java
public String getIme() {
    return ime;
}

public void setIme(String ime) {
    this.ime = ime;
}
```

```java
public String vrniIme() {
    return ime;
}

public void nastaviIme(String ime) {
    this.ime = ime;
}
```

```java
if (datumRojstva == null)
    System.out.println("Je null");
else System.out.println("Ni null");
```

```java
if (datumRojstva == null) {
    System.out.println("Je null");
} else {
    System.out.println("Ni null");
}
```

```java
if (datumRojstva == null)
{
    System.out.println("Je null");
} else {
    System.out.println("Ni null");
}
```

```
private static SimpleDateFormat sdf=new SimpleDateFormat("dd. MM. yyyy HH:mm:ss");
@Override
public String toString() {
    return ime + " " + priimek + ","+spol+" ("+email+"); datum rojstva:
    "+sdf.format(datumRojstva.getTime())+"; vpis: "+sdf.format(datumVpisa.getTime());
}

@Override
public String toString() {
    SimpleDateFormat sdf=new SimpleDateFormat("dd. MM. yyyy HH:mm:ss");
    return ime + " " + priimek + ","+spol+" ("+email+"); datum rojstva:„
    +sdf.format(datumRojstva.getTime())+"; vpis: "+sdf.format(datumVpisa.getTime());
}

@Override
public String toString()
{
    SimpleDateFormat sdf=new SimpleDateFormat("dd. MM. yyyy HH:mm:ss");
    return ime + " " + priimek + ","+spol+" ("+email+"); datum rojstva:„
    +sdf.format(datumRojstva.getTime())+"; vpis: "+sdf.format(datumVpisa.getTime());
}
```

```java
@Override
public List<Oseba> vrniVse(Connection conn) throws Exception {
    List<Oseba> ret = new ArrayList<Oseba>();
    ResultSet rs=conn.createStatement().executeQuery("select * from oseba");
    while (rs.next()) {
        Oseba o = new Oseba(rs.getString("ime"), rs.getString("priimek"));
        o.setId(rs.getInt("id"));
        o.setSpol(rs.getString("spol"));
        o.getDatumVpisa().setTimeInMillis(rs.getTimestamp("cas").getTime());
        o.getDatumRojstva().setTimeInMillis(rs.getTimestamp("rojstvo").getTime());
        o.setKontakti(KontaktDao.getInstance().vrniVse(o.getId(),conn));
        ret.add(o);
    }
    rs.close();
    return ret;
}
```

```java
@Override
public List<Oseba> vrniVse(Connection conn) throws Exception {
    log.info("Vračam vse osebe!");
    List<Oseba> ret = new ArrayList<Oseba>();
    ResultSet rs=conn.createStatement().executeQuery("select * from oseba");
    while (rs.next())
    {
        Oseba o = new Oseba(rs.getString("ime"), rs.getString("priimek"));
        o.setId(rs.getInt("id"));
        o.setSpol(rs.getString("spol"));
        o.getDatumVpisa().setTimeInMillis(rs.getTimestamp("cas").getTime());
        o.getDatumRojstva().setTimeInMillis(rs.getTimestamp("rojstvo").getTime());
        o.setKontakti(KontaktDao.getInstance().vrniVse(o.getId(),conn));
        ret.add(o);
    }
    rs.close();
    return ret;
}
```

```java
@Override
public List<Oseba> vrniVse(Connection conn) throws Exception {
    List<Oseba> ret = new ArrayList<Oseba>();
    try {
        log.info("Vračam vse osebe!");
        ResultSet rs=conn.createStatement().executeQuery("select * from oseba");
        while (rs.next()) {
            Oseba o = new Oseba(rs.getString("ime"), rs.getString("priimek"));
            o.setId(rs.getInt("id"));
            o.setSpol(rs.getString("spol"));
            o.getDatumVpisa().setTimeInMillis(rs.getTimestamp("cas").getTime());
            o.getDatumRojstva().setTimeInMillis(rs.getTimestamp("rojstvo").getTime
            ());
            o.setKontakti(KontaktDao.getInstance().vrniVse(o.getId(),conn));
            ret.add(o);
        }
        rs.close();
    } catch (Exception e) {
        e.printStackTrace();
    }
    return ret;
}
```

```java
@Override
public List<Oseba> vrniVse(Connection conn) {
    List<Oseba> ret = new ArrayList<Oseba>();
    try {
        log.info("Vračam vse osebe!");
        ResultSet rs=conn.createStatement().executeQuery("select * from oseba");
        while (rs.next()) {
            Oseba o = new Oseba(rs.getString("ime"), rs.getString("priimek"));
            o.setId(rs.getInt("id"));
            o.setSpol(rs.getString("spol"));
            o.getDatumVpisa().setTimeInMillis(rs.getTimestamp("cas").getTime());
            o.getDatumRojstva().setTimeInMillis(rs.getTimestamp("rojstvo").getTime());
            o.setKontakti(KontaktDao.getInstance().vrniVse(o.getId(),conn));
            ret.add(o);
        }
        rs.close();
    } catch (Exception e) {
        log.error(e.getMessage());
    }
    return ret;
}
```

```
<tr>
    <td>Ime:</td>
    <td><h:inputText value="#{osebe.novaOseba.ime}"/></td>
</tr>
```

```
<tr>
    <td>
        Ime:
    </td>
    <td>
        <h:inputText value="#{osebe.novaOseba.ime}"></h:inputText>
    </td>
</tr>
```

BancniRacun

BancniRacunVAO

BancniRacunImpl

si.um.feri.ris.vao.BancniRacun

si.um.feri.ris.mobilna.BancniRacun

```
//tale metoda ne deluje najbolje. do-not-touch
public void BancniRacun(String stevilka, double znesekZaNakazilo) {
  . . .
}
```

```
/*
set the value of the age integer to 32
*/
int age = 32;
```

```
/*
Class used to workaround Richard being a f***ing idiot
*/
```

Dokumentacija, dokumentacija…

Standardi kodiranja

Pravila dela v skupini

Navodila za razvijalce

Stilni vodiči

…

Skupinsko delo!

Skupno lastništvo kode!

Standardi kodiranja (Code standard) #1/2

Namen: Zanesljiva koda, ki jo je enostavno (možno) vzdrževati

Skupinsko delo, skupinsko lastništvo kode
    Pa tudi če ne – vzdrževanje!

Nabor dobrih / zaželenih / obveznih praks

Vezani na programski jezik

1. **Splošno** uveljavljeni
2. Vezano na **podjetje**
3. Specifične zahteve za **projekt**

Tudi za druga področja
    Npr. stilni vodiči

Standardi kodiranja (Code standard) #2/2

> *"Establish programming conventions before you begin programming. It's nearly impossible to change code to match them later." [McConnell]*

40% – 80% stroškov življenjskega cikla programske opreme je v vzdrževanju

Le redko vzdržujete programsko opremo, ki ste jo napisali sami

Izboljšajo berljivost in razumevanje kode

Če je izvorna koda vaš produkt, je še posebej pomembno, da sledite smernicam

https://firefox-source-docs.mozilla.org/code-quality/coding-style/index.html

Standard kodiranja

Ni treba, da je dolg!
- Kratko, jedrnato, razumljivo
- Vzpodbujamo „očitne“ (ZKP ☺) prakse

Vsebina:
- Poimenovanja, lokacije datotek
- Poimenovanja razredov, metod, spremenljivk
- Način gnezdenja
- Način obravnave izjem
- Namen / način / lokacije komentarjev
- Uporaba konstruktov jezika
- Zahteve večjezičnosti / monitoringa / analitike …
- Uporabljen (naravni) jezik
- …
- Ostale podrobnosti

Navodila za razvijalce

Lahko tudi v obliki prototipne implementacije!

https://gitlab.com/luka.pavlic/bestmeddev/-/tree/master/microservices/prototype

Podpora orodij!

Bodoči sodelavci na projektu
- Tudi vzdrževalci ipd.
- Vzpostavitev razvojnih okolij

https://www.atlassian.com/software/crucible

Crucible

Razvojna okolja, različice

Knjižnice, različice

Način vzpostavitve razvojnega okolja

Povzetek poglavitnih delov projekta (projektov)

https://www.sonarsource.com/products/sonarlint/

sonarlint

Včasih orodja in knjižnice vključimo tudi v sistem za upravljanje konfiguracije

https://prettier.io/

Vzdržujemo sproti!

CODACY

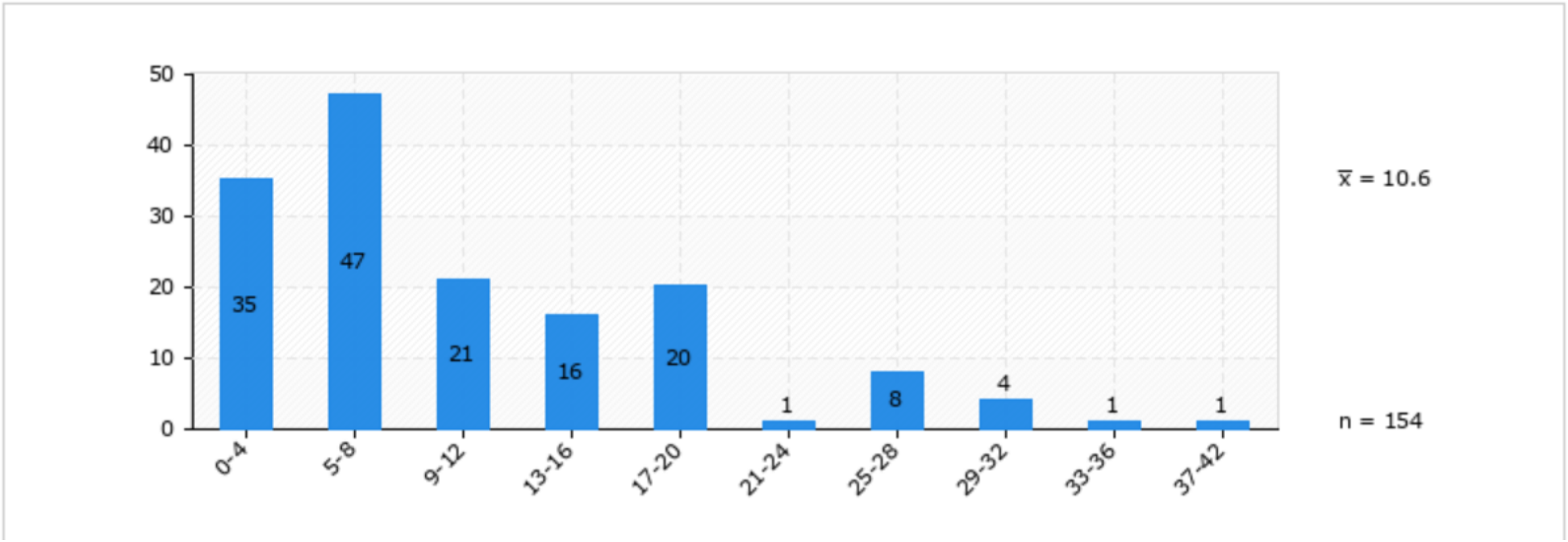
https://www.codacy.com

ESLint

https://eslint.org

Slovenska podjetja; Vzorec: 154; Uporaba standardov kodiranja v praksi

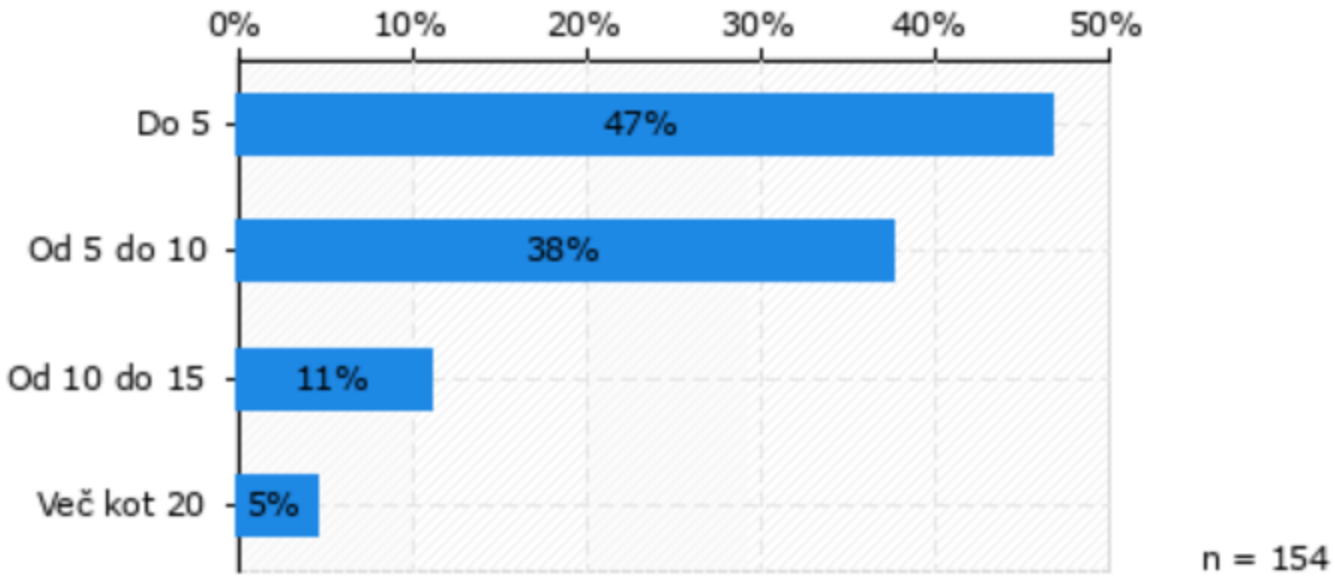
Koliko let izkušenj imate z razvojem informacijskih rešitev? (n = 154)



Kakšno je vaša prevladujoča vloga pri projektih razvoja informacijskih rešitev? (n = 154)

Koliko zaposlenih ima vaše podjetje? (n = 154)

Koliko članov običajno šteje vaša ekipa? (n = 154)



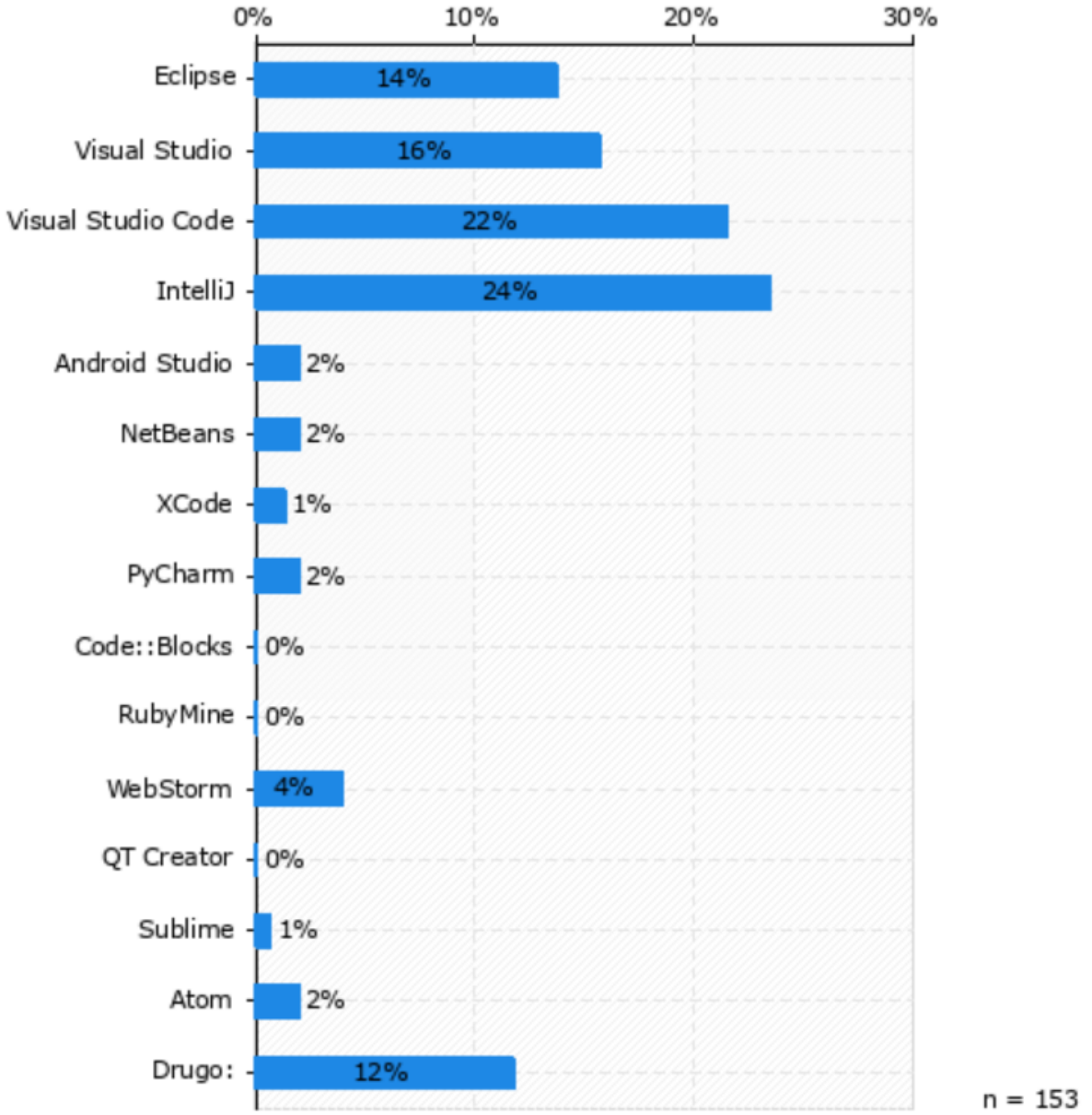
Slo

sta

Izvorna ko

Na

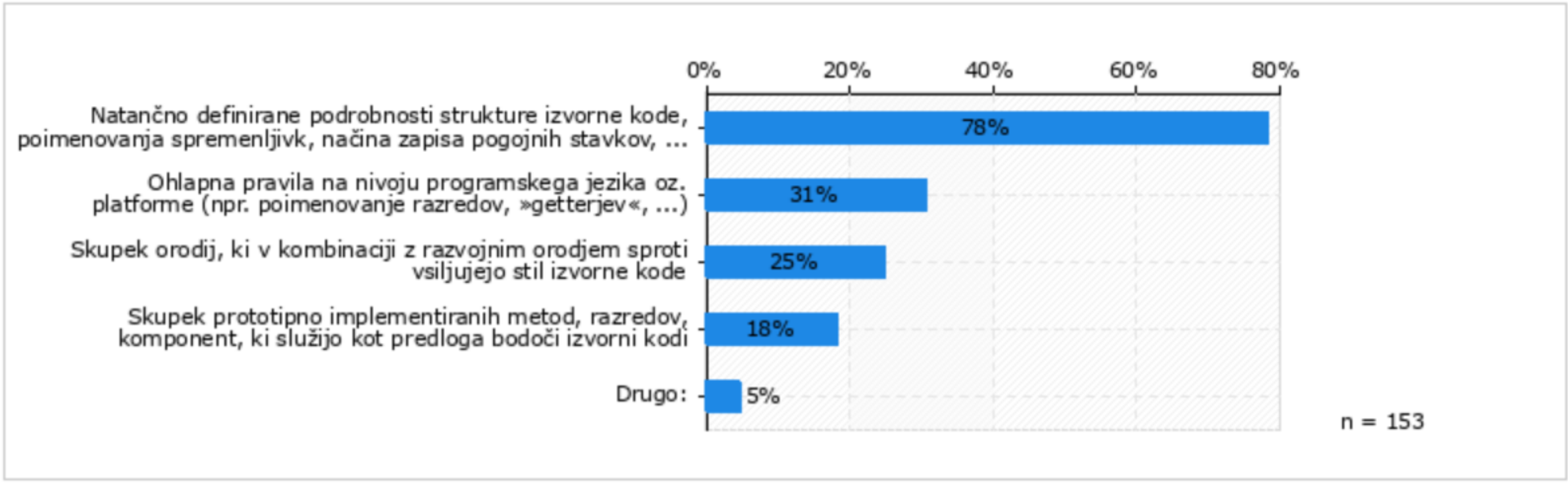
Katero je vaše prevladujoče razvojno okolje, s katerim se srečate pri razvoju informacijskih rešitev? (n = 153)

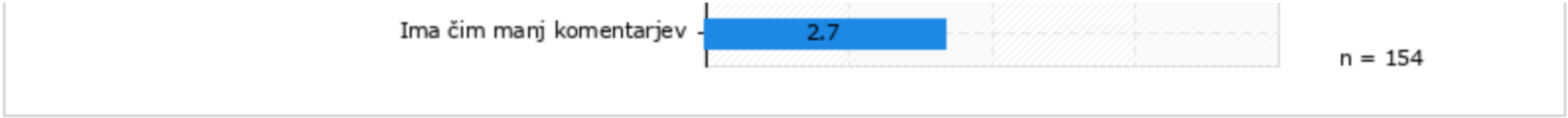


Slovenska podjetja: Vzorec: 154; Uporaba

Kaj za vas pomeni izraz »standard kodiranja« ? (n = 153)

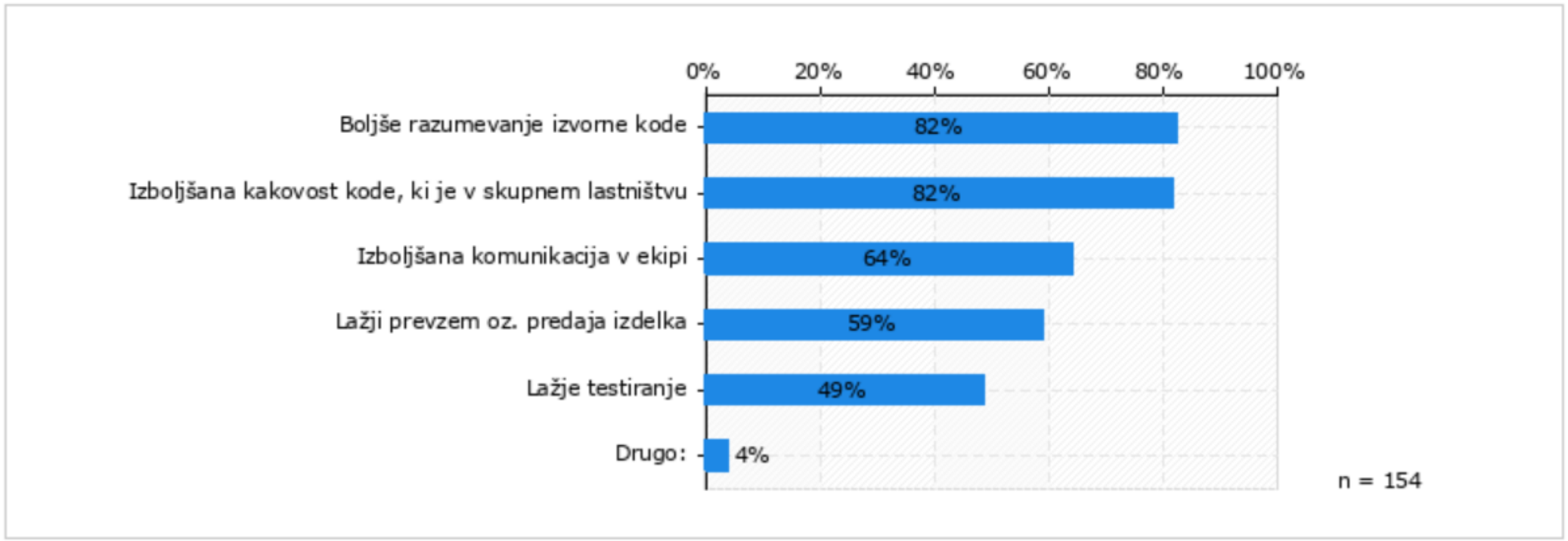
Možnih je več odgovorov





Katere menite, da so koristi uporabe standardov kodiranja? (n = 154)

Možnih je več odgovorov

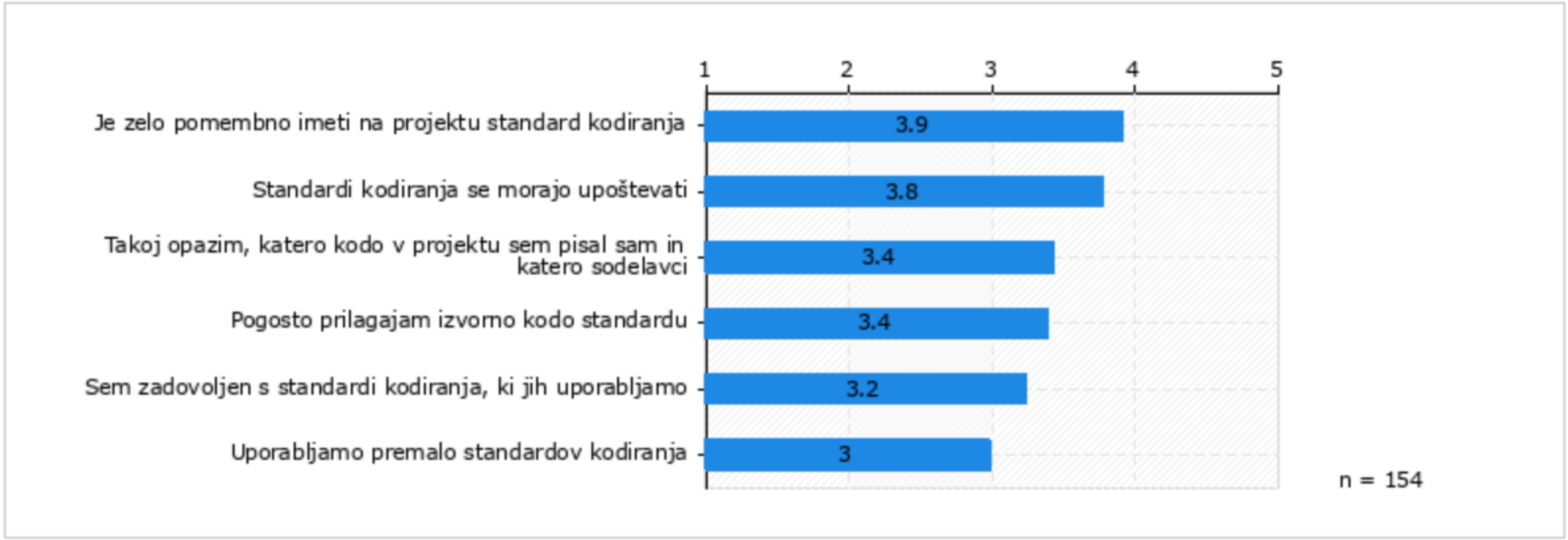


Q15 (Drugo:)

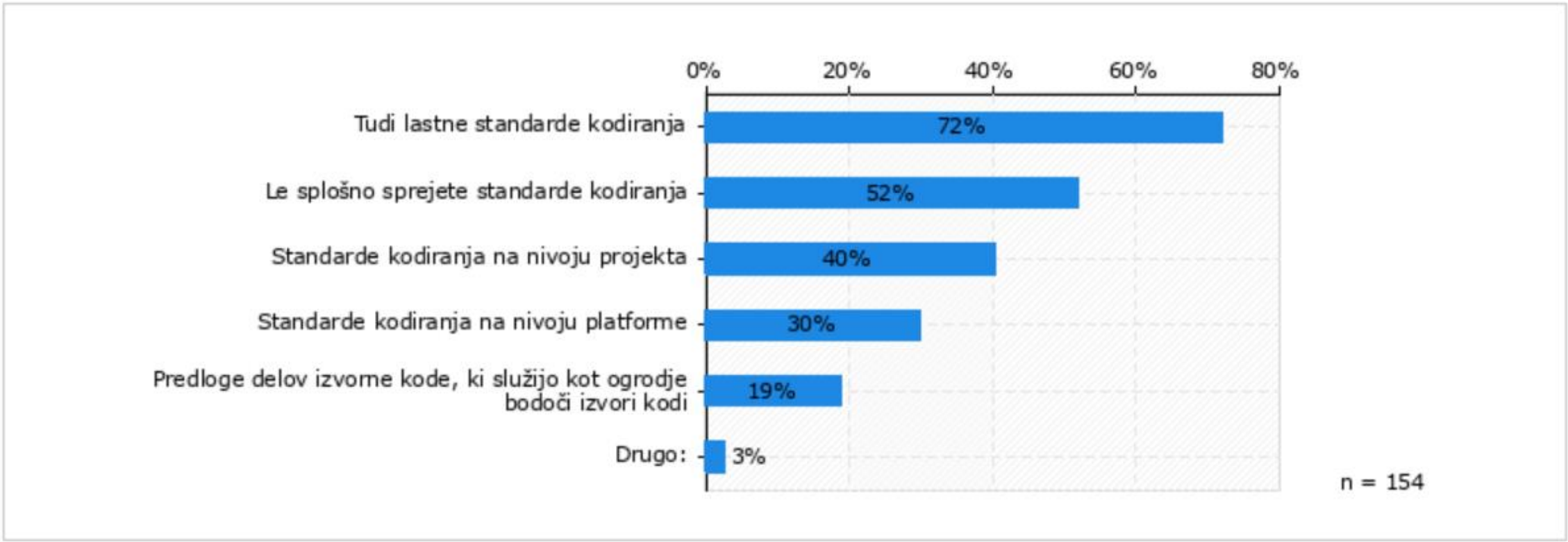
hitrejši onboarding na projekt
lažji onboarding juniorjev
zadostitev pogojev stranke
v kodi, ki je razvijalec ni sam napisal: lažje odkrivanje napak, lažje dodajanje novih funkcionalnosti
lažja berljivost kode, lažje vzdrzevanje, bolj kakovostna koda
lažje vzdrževati dolgoročno

Slovenska podjetja; Vzorec: 154; Uporaba standardov kodiranja v praksi

Opažam da: (n = 154)



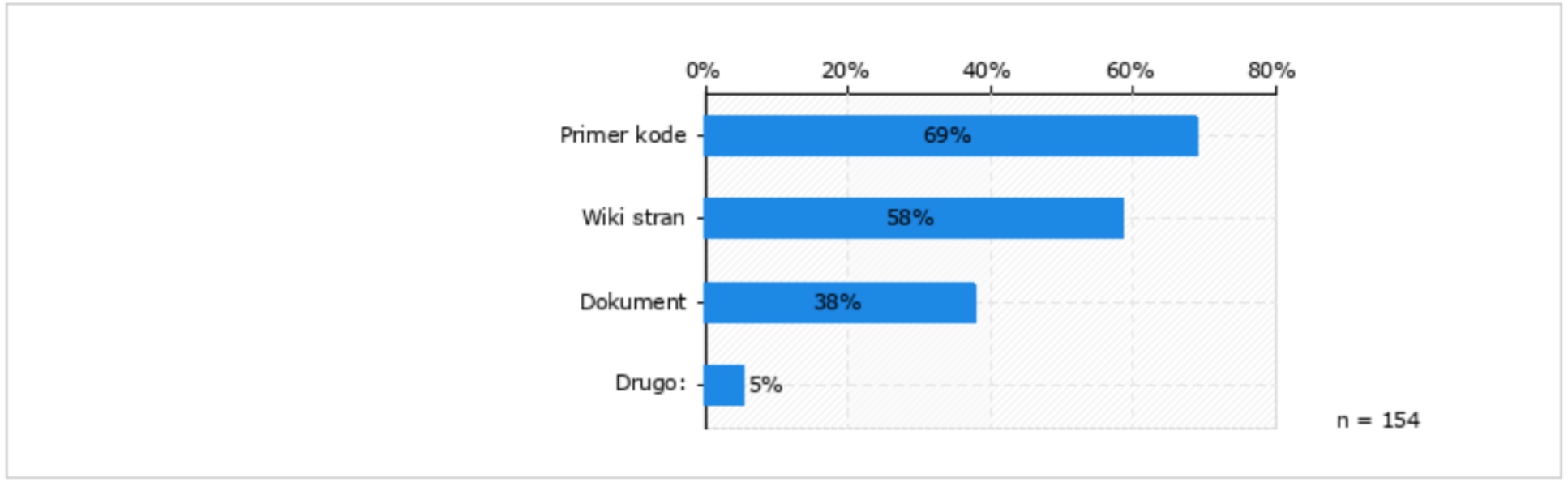
Pri delu uporabljamo: (n = 154)
Možnih je več odgovorov



Q17 (Drugo:)

včsih je vezano nz zahteve naročnika
standarde kodiranja na nivoju produkta, ki se razvija
kak se zmenis s sodelovci. simple as that
ničesar

Kakšno obliko dokumentiranja standardov kodiranja si želite? (n = 154)
Možnih je več odgovorov

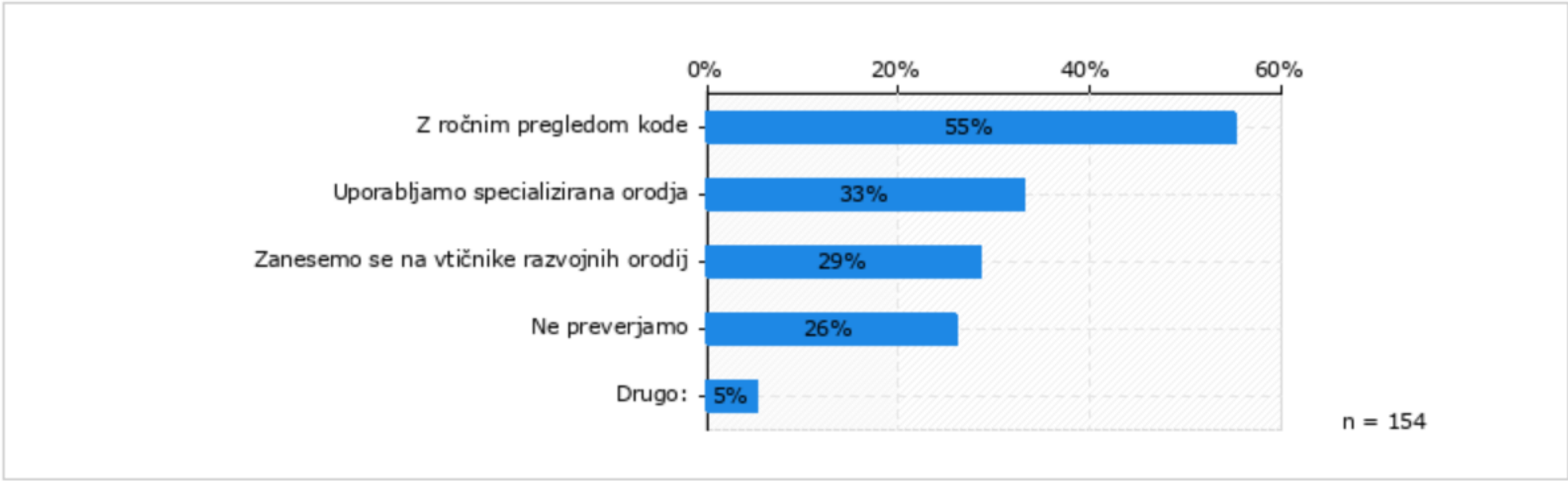


Q19 (Drugo:)

markdown dokumenti
avtomatsko preko toolinga - e.g .editorconfig
oralni dogovor
ide template
postman
razvojno orodje, ki uporabnika sili v določeno strukturo
karkoli, kjer hitro prideš do informacij
linter

Slovenska podjetja; Vzorec: 154; Uporaba standardov kodiranja v praksi

Na kakšen način preverjate, da se standardi upoštevajo? (n = 154)
Možnih je več odgovorov



Ko dokumentacija niso dokumenti

https://github.com/mihaprah/projekt

https://gitlab.com/luka.pavlic/bestmeddev

https://github.com/lukapavlic/ots-prijave

https://gitlab.com/luka.pavlic/robobum/-/tree/main/backend?ref_type=heads

https://github.com/lukapavlic/primeri/tree/master/spring_measurements_backend

https://github.com/lukapavlic/NanoCLM/tree/main/CLMRepo

https://docs.github.com/en/repositories/managing-your-repositorys-settings-and-features/customizing-your-repository/about-readmes

README.md – primer vsebine

1. [About the Project](#about-the-project)
2. [Technologies Used](#technologies-used)
3. [Project Structure](#project-structure)
4. [Getting Started](#getting-started)
   - [Prerequisites](#prerequisites)
   - [Installation](#installation)
   - [Environment Variables](#environment-variables)
5. [Running the App](#running-the-app)
   - [Running Frontend](#running-frontend)
   - [Running Backend](#running-backend)
6. [Testing](#testing)
7. [Deployment](#deployment)
   - [Deploying Frontend](#deploying-frontend)
   - [Deploying Backend](#deploying-backend)
8. [Contributing](#contributing)
9. [License](#license)

1. *Splošen pregled projekta in njegovih ciljev.*
2. *Seznam orodij, knjižnic in ogrodij, uporabljenih v frontend in backend delu.*
3. *Pregled strukture direktorijev (npr. za frontend in backend).*
4. *Koraki za nastavitev razvojnega okolja, vključno s predpogoji, namestitvijo in okoljskimi spremenljivkami.*
5. *Navodila za zagon aplikacij.*
6. *Navodila za izvajanje testov.*
7. *Koraki za namestitev frontend in backend aplikacij.*
8. *Smernice za prispevanje k projektu.*
9. *Informacije o licenci projekta.*

Ali vsaj:

```
# Project Title
A short description of your project goes here.

## Table of Contents
- [Installation](#installation)
- [Usage](#usage)
- [Contributing](#contributing)
- [License](#license)
```