کنفرانس بین‌المللی در مورد نرم افزار و فرآیندهای سیستم

به سمت یک انبار دانش و سیستم خبره برای خودکار کردن وظایف **SDLC(چرخه حیات تولید نرم افزار)**

ریتو کاپور

گروه علوم کامپیوتر و مهندسی موسسه فناوری هند روپار

ایمیل: [ritu.kapur@iitrpr.ac.in](mailto:ritu.kapur@iitrpr.ac.in)

تحت نظارت Balwinder Sodhi

گروه علوم کامپیوتر و مهندسی موسسه فناوری هند روپار

ایمیل: [sodhi@iitrpr.ac.in](mailto:sodhi@iitrpr.ac.in)

**چکیده: هزینه یک توسعه دهنده نرم­افزار ماهر و شایسته زیاد است و مطلوب است که وابستگی به این منابع انسانی پرهزینه را به حداقل برسانید. یکی از راه­های به حداقل رساندن چنین هزینه­هایی خودکار کردن کارهای مختلف توسعه نرم افزار است.**

**پیشرفت­های اخیر در هوش مصنوعی (AI) و در اختیار داشتن حجم وسیعی از داده­های دارای دانش در مکان­های مختلف مرتبط با توسعه نرم­افزار، فرصتی مناسب برای ساختن ابزارهایی است که می­توانند کارهای توسعه نرم­افزار را خودکار کنند. برای مثال، دانش پنهان قابل‌ توجهی در داده‌های خام یا بدون ساختار مرتبط با اقلامی مانند فایل‌های منبع، کد نویسی، گزارش‌های خرابی، توضیحات و غیره وجود دارد که در مخازن نرم افزار منبع باز (OSS) موجود است.**

**هدف ما این است که از چنین داده­های دانش بنیان، آخرین پیشرفت­های هوش مصنوعی و سخت­افزاری، برای ایجاد انبارهای دانش و سیستم های خبره برای حوزه توسعه نرم­افزار بهره ببریم. چنین ابزارهایی می‌توانند به ایجاد برنامه‌های کاربردی برای انجام کارهای مختلف توسعه نرم‌افزار مانند: پیش‌بینی عیب، تخمین تلاش، بازبینی کد و غیره کمک کنند.**

**واژه‌های کلیدی - مهندسی نرم‌افزار خودکار، نگهداری نرم‌افزار، داده‌کاوی، یادگیری نظارت شده**

1. انگیزه

**موفقیت یک پروژه توسعه نرم­افزار تا حد زیادی به مهارت و تجربه مهندسان نرم­افزار که در حال کار روی پروژه هستند بستگی دارد. علاوه بر این، هزینه مهندسین ماهر به بخش عمده‌ای از هزینه کلی پروژه کمک می‌کند. بنابراین، کاهش وابستگی به چنین منابع مهارتی بسیار مطلوب و مؤثرتر است. یکی از راه‌های دستیابی به چنین هدفی، ساخت ابزارها و تکنیک‌هایی است که می‌توانند به طور کامل یا بخشی از وظایف توسعه نرم‌افزار را خودکار کنند و این تمرکز اصلی کار ماست.**

**الف) اصول اساسی سیستم ما**

**خلاصه آن می‌تواند به شکل فرضیه تحقیق و مراحل تحقیق مشخص‌شده در این بخش بیان شود .**

1. **فرضیه : فرضیه‌ای که ما با آن کار می‌کنیم به شرح زیر است :**

**آ) توسعه نرم‌افزار یک فرآیند خلاقانه است و نتایج یک کار توسعه نرم‌افزار به مهارت مهندسین بستگی دارد.**

**ب) اطلاعات مفیدی درباره تکمیل و توسعه نرم‌افزار وجود دارد. این اطلاعات می‌تواند برای ساخت ابزارها و تکنیک‌های خودکارسازی وظایف توسعه نرم‌افزار به کار گرفته شود، در نتیجه به کاهش هزینه‌ها کمک می‌کند. به عنوان مثال ، پروژه های منبع آزاد موجود در GitHub، مباحث مربوط به برنامه نویسان در StackOverﬂow و غیره منابع غنی از داده­ها و مصنوعات دیگری را ارائه می دهند که می توان از آن­ها استفاده کرد.**

**2) مراحل تحقیق: مراحل اصلی تحقیق برای دستیابی به اهداف ذکر شده در بند قبل را می توان به شرح زیر ذکر کرد:**

**آ) منابعی را شناسایی كنید كه درآن می­توانیم اطلاعات مفید در مورد وظایف توسعه نرم­افزار در پروژه­های تکمیل شده یا فعال را استخراج کنیم.**

**ب) اطلاعات مربوطه را از منابع فوق استخراج كنید كه می­تواند هنگام تهیه­ی نرم­افزار، روند تصمیم گیری را راهنمایی كند. تصمیمات در اینجا شامل تصمیمات فنی و مدیریتی هستند. برای مثال، تخمین تلاش توسعه، شناسایی و مکان‌یابی عیوب و غیره.**

**ج) با استفاده از اطلاعات فوق، ابزارهای مناسب مبتنی بر هوش مصنوعی را برای دستیابی به اهداف فوق بسازید.**

**تصمیمات طراحی نرم‌افزار ضعیف اغلب منجر به یک نرم‌افزار کیفیت ضعیف می‌شود که منجر به هزینه نگهداری بالا می‌شود، در حالی که یک برآورد تلاش توسعه نرم‌افزار ضعیف می‌تواند سبب زیان قابل‌توجهی برای توسعه دهنده شود. بنابراین مطلوب است که ابزارها و تکنیک­های متنوعی ساخته شود که بتواند در چنین فعالیت­های تصمیم گیری کمک کند و عدم قطعیت را کاهش یا احتمالاً از بین ببرد.**

**ب) هر نرم‌افزار از عناصر نرم‌افزاری پایه تشکیل شده‌است.**

**یک نرم­افزار از عناصر اساسی خاصی مانند کامپوننت، کانکتور، چارچوب، کتابخانه و غیره تشکیل شده است. بنابراین، تلاش برای توسعه نرم­افزار را می­توان با اضافه کردن تلاش صرف شده برای ساختن عناصر اصلی نرم افزار آن محاسبه کرد.**