เทคนิคสำหรับการวัดคุณภาพของโค้ดเชิงปริมาณที่มีอยู่หลากหลายจะมีการใช้ชุดของมาตรวัดที่แตกต่างกันและมักก่อให้เกิดความไม่สอดคล้องกันและ/หรือไม่สามารถเปรียบเทียบค่าคุณภาพของแต่ละชุดได้ ชุดค่าดัชนีเหล่านี้แสดงส่วนประกอบของผลลัพธ์ที่ประเมินคุณภาพของโค้ดจากหลากหลายมุมมองซึ่งค่าแต่ละค่าไม่จำเป็นต้องสอดคล้องกัน ดังนั้นเมื่อมีโค้ดมากกว่าหนึ่งสำหรับลักษณะการทำงานของซอฟต์แวร์อย่างใดอย่างหนึ่งแล้วนั้นการเลือกโค้ดที่ดีกว่าจึงเป็นความท้าทาย งานวิจัยนี้เสนอว่าสามารถนำเทคนิคกระบวนการลำดับชั้นเชิงวิเคราะห์มาใช้เป็นเครื่องมืออันสะดวกสำหรับช่วยผู้พัฒนาอย่างมีประสิทธิผลในการพิจารณาเลือกโค้ดที่เหมาะสมที่สุดสำหรับเป้าหมายการออกแบบที่กำหนด แนวทางที่นำเสนอจะคำนวณค่าลักษณะประจำคุณภาพสำหรับโค้ดที่กำหนด จากนั้นประยุกต์ใช้กระบวนการลำดับชั้นเชิงวิเคราะห์เพื่อจัดลำดับลักษณะประจำคุณภาพของโค้ด โดยการเปรียบเทียบความสำคัญเป็นรายคู่และหาค่าดัชนีคุณภาพที่แสดงถึงระดับคุณภาพของโค้ดนั้นเป็นลำดับต่อมา การอภิปรายผลการตรวจสอบกระบวนวิธีตามที่กำหนดไว้สามารถสรุปได้ว่ากระบวนการลำดับชั้นเชิงวิเคราะห์เป็นเครื่องมือที่มีประสิทธิภาพและมีความเหมาะสมสำหรับการเปรียบเทียบคุณภาพของโค้ด

228088

Techniques for a quantitative measurement of code quality diversely exist that utilize different sets of metrics and that often lead to non-compatible and/or non-comparable sets of quality values. A set of these quality values represents the composite result that evaluates the various aspects of code quality not necessarily in agreement with each other. Consequently, when several implementations exist for a particular software feature, choosing one that is better could be challenging. This thesis proposes that the analytic hierarchy process (AHP) technique can provide a convenient means to effectively assist software developers in rationalizing among candidate implementations to find the one that is best suited for the design goal. The proposed method computes quality attributes for given source code. It then applies AHP against the ranked quality attributes, in a pair-wise manner, to derive the associated priority factors, and, subsequently, the quality index that indicates the quality level of the source code. Discussion on the validity of the proposed method follows that leads to a conclusion that AHP is indeed an effective and convenient tool for comparing code quality.